

TUGAS AKHIR - KI141502

REKOMENDASI DESAIN *UX (USER EXPERIENCE)* PADA APLIKASI DRUM VIRTUAL BERBASIS TEKNOLOGI INTEL REALSENSE

DIDIT SEPIYANTO
NRP 5113100090

Dosen Pembimbing
Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Anny Yuniarti, S.Kom., M.Comp.Sc.

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017



TUGAS AKHIR - KI141502

REKOMENDASI DESAIN *UX* (*USER EXPERIENCE*) PADA APLIKASI DRUM VIRTUAL BERBASIS TEKNOLOGI INTEL REALSENSE

**DIDIT SEPIYANTO
NRP 5113100090**

**Dosen Pembimbing
Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Anny Yuniarti, S.Kom., M.Comp.Sc.**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**

(Halaman ini sengaja dikosongkan)



FINAL PROJECT- KI141502

**UX (USER EXPERIENCE) DESIGN
RECOMMENDATION IN VIRTUAL DRUM
APPLICATION BASED TECHNOLOGY OF INTEL
REALSENSE**

**DIDIT SEPIYANTO
NRP 5113100090**

**Advisor
Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Anny Yuniarti, S.Kom., M.Comp.Sc.**

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2017**

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

REKOMENDASI DESAIN UX (USER EXPERIENCE) PADA APLIKASI DRUM VIRTUAL BERBASIS TEKNOLOGI INTEL REALSENSE

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Rumpun Mata Kuliah Interaksi, Grafika, dan Seni
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

DIDIT SEPIYANTO

NRP. 5113 100 090

Disetujui oleh Dosen Pembimbing Tugas Akhir

Wijayanti Nurul Khotimah, S.Kom

NIP: 198603122012122004

Anny Yuniarti, S.Kom., M.Comp.Sc.

NIP: 198106222005012002



(pembimbing 1)

(pembimbing 2)

**SURABAYA
JANUARI, 2017**

(Halaman ini sengaja dikosongkan)

REKOMENDASI DESAIN UX (USER EXPERIENCE) PADA APLIKASI DRUM VIRTUAL BERBASIS TEKNOLOGI INTEL REALSENSE

Nama Mahasiswa : Didit Sepiyanto
NRP : 5113100090
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Wijayanti Nurul Khotimah, S.Kom., M.Sc
Dosen Pembimbing II : Anny Yuniarti, S.Kom., M.Comp.Sc.

ABSTRAK

Musik adalah suara yang disusun demikian rupa sehingga mengandung irama, lagu, dan keharmonisan terutama suara yang dihasilkan dari alat yang dapat menghasilkan bunyi-bunyian. Terdapat beberapa jenis alat musik berdasarkan cara memainkannya salah satunya yaitu alat musik pukul drum. Harga alat musik ini cukup mahal sekitar 13 juta dalam satu setnya. Selain itu, dibutuhkan ruangan yang lebih luas untuk meletakkan alat musik ini.

Intel Realsense 3D Camera adalah teknologi kamera yang mampu merespon tangan, lengan, dan gerakan kepala serta ekspresi wajah. Kemampuan kamera Intel Realsense ini dapat mendeteksi 22 sendi di satu telapak tangan kita dan juga dapat mendeteksi kedalaman objek secara 3 Dimensi.

Intel Realsense adalah pengembangan teknologi dari HCI (Human-Computer Interaction) dimana di dalam HCI itu sendiri terdapat berbagai cabangnya diantaranya UI (User Interface) dan UX (User Experience). UX memiliki ranah yang lebih luas dari UI, karena ranah UX ini dimulai dengan research pasar pengguna yang kemudian diimplementasi kedalam sebuah interface.

Tugas akhir ini mencoba untuk menerapkan desain UX pada aplikasi drum virtual menggunakan teknologi Intel Realsense. Teknologi Intel Realsense ini, memungkinkan pengguna dapat membunyikan suara dengan cara menggerakkan tangan

seperti memukul drum pada kamera Intel Realsense. Tugas akhir ini diharapkan dapat memberikan pengalaman berbeda bagi pengguna saat memainkan aplikasi drum virtual dengan cara menggerakkan tangan.

Terdapat 3 tipe desain UI/UX (Desain tipe A, B, C) aplikasi drum virtual yang akan dibuat. Dari ketiga desain tersebut akan dipilih desain mana yang sesuai untuk aplikasi drum virtual menggunakan Intel Realsense. Dari hasil pengujian dengan cara melakukan experiment task dan melakukan beberapa kuesioner desain yang baik untuk untuk pembuatan aplikasi drum virtual adalah desain tipe C sedangkan untuk hasil dependent variable seperti task complete, task accuracy dan task error adalah desain tipe A.

Kata kunci: Drum, Intel Realsense, Musik, User Interface, User Experience, Virtual Drum.

UX (USER EXPERIENCE) DESIGN RECOMMENDATION IN VIRTUAL DRUM APPLICATION BASED TECHNOLOGY OF INTEL REALSENSE

Student Name : Didit Sepiyanto
NRP : 5113100090
Major : Teknik Informatika FTIf-ITS
Advisor I : Wijayanti Nurul Khotimah, S.Kom., M.Sc.
Advisor II : Anny Yuniarti, S.Kom., M.Comp.Sc.

ABSTRACT

Music is organized sound such that it contains rhythm, song, and harmony particularly sound generated from a tool that can generate sounds. There are several kinds of musical instruments is based on how to play, one of them is percussion drum. The price of this instrument is quite expensive at around 13 millio. In addition, it takes more space to put this instrument.

Intel Realsense 3D Camera is a camera technology that is capable of responding to the hand, arm, and head movements and facial expressions. Realsense camera capabilities Intel is able to detect 22 joints in the palm of our hands and also can detect the depth of objects in 3 Dimensions.

Intel Realsense is developing the technology of HCI (Human-Computer Interaction) where there are many branches of which UI (User Interface) and UX (User Experience). UX has a wider sphere of UI, because the realm UX starts with market research users are then implemented into a single interface.

The final task is to try to implement a UX design on a virtual drum applications using Intel technology Realsense. Intel's Tteknologi Realsense, allows users to mute by way of moving the hand like drumming on Intel Realsense camera. The final project is expected to provide a different experience for users when playing a virtual drum application by way of moving the hand.

There are 3 types of design UI / UX (Design type A, B, C) application of virtual drums to be made. Of the three designs will

be selected designs which are suitable for applications using Intel Realsense virtual drum. From the test results by doing experimental task and do some good for the questionnaire design for manufacturing applications is a virtual drum design type C while the dependent variable outcomes such as task complete, task accuracy and error task is the design of type A.

Key words: *Drum, Intel Realsense, Music, User Interface, User Experience, Virtual Drum.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur ke hadirat Allah Subhanahuwata'ala yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “Rekomendasi Desain UX (*User Experience*) pada Aplikasi Drum Virtual Berbasis Teknologi Intel Realsense”.

Pengerjaan tugas akhir ini adalah syarat bagi penulis untuk mengeluarkan seluruh ilmu yang telah didapat semenjak di awal perkuliahan sampai hari ini di lingkungan jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya

Dalam pelaksanaan dan pembuatan tugas akhir ini, penulis menerima banyak sekali bantuan dari berbagai pihak, baik secara moril maupun materi. Melalui lembaran kata pengantar ini, penulis secara khusus menyampaikan ucapan terima kasih kepada:

1. Allah Subhanahu wa ta'ala yang telah melimpahkan kesehatan, semangat, dan hidayah-Nya sehingga penulis Alhamdulillah mampu menyelesaikan tugas akhir dengan baik.
2. Junjungan kita Nabi Muhammad SAW yang telah menjadi inspirasi, contoh yang baik bagi penulis sehingga tetap termotivasi dalam mengerjakan tugas akhir.
3. Kedua orang tua penulis yang telah mencurahkan doa, dukungan semangat, perhatian, serta kasih sayang kepada penulis.
4. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku kepala jurusan Teknik Informatika ITS.

5. Ibu Bilqis Amaliah, S.Kom.,M.Kom. selaku dosen wali yang senantiasa memberikan arahan.
6. Ibu Wijayanti Nurul Khootimah, S.Kom., M.Sc. dan Ibu Anny Yuniarti, S.Kom., M.Comp.Sc. selaku dosen pembimbing yang senantiasa memberikan arahan.
7. *Search engine* Google yang melimpahkan penulis terhadap informasi-informasi penting terkait pembuatan aplikasi ini.
8. Novita Retno, yang sudah membantu memberi semangat dan pembuatan objek-objek, sehingga bisa penulis pakai untuk lanjutan pembuatan aplikasi ini.
9. Prasetya Gilang Nuswantara, Ibnu Prayogi, Relaci Aprilia, Mardiana Sekarsari, Burhanuddin Rasyid, Hanif Sudira, Usaid Shidqul Chaq, Humaira Try Acantya, Petrus Damianus Sammy Wiyadi, Elva Maulidiah, Ivaldy Putra serta teman-teman sesama dalam perjuangan di laboratorium Rekayasa Perangkat Lunak, telah memberikan arahan dan semangat dalam pengerjaan tugas akhir ini.
10. Teman-teman seangkatan yang selalu siap sedia ketika penulis mengalami kesulitan.
11. Teman-teman angkatan 2013, 2014 dan 2015 yang sudah memberikan pengalaman selama kuliah di Teknik Informatika ini.
12. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya tugas akhir ini, penulis mengucapkan terima kasih.

Penulis telah mengerjakan sebaik mungkin dalam penyusunan aplikasi tugas akhir ini. Penulis juga memohon maaf apabila terdapat kekurangan, kesalahan, maupun kelalaian yang telah penulis lakukan.

Surabaya, 21 Januari 2017
Penulis

Didit Sepiyanto

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xii
DAFTAR ISI	xv
DAFTAR GAMBAR	xx
DAFTAR TABEL	xxii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Pembuatan Tugas Akhir	2
1.5 Manfaat	2
1.6 Metodologi.....	3
1.7 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Intel Realsense	7
2.2 Desain <i>UI (User Interface)</i>	11
2.3 Desain <i>UX (User Experience)</i>	13
2.4 Musik	14
2.5 Alat Musik Drum.....	15
2.6 Not Balok Drum	18
2.7 Unity	20
2.8 Diagram <i>Use Case</i>	20
BAB III ANALISIS DAN PERANCANGAN	23
3.1 Analisis Sistem	23
3.2 Perancangan <i>UI (User Interface)</i> Alat Musik Drum	24
3.3 Perancangan Proses	24
3.3.1 Rancangan Proses Pendeteksi Sendi pada Jari Tangan	24
3.3.2 Rancangan Proses Pemutaran Suara Saat Tangan Menyentuh Drum.....	25

3.3.3 Rancangan Proses Pendeteksi <i>Gesture</i> (Pola Tangan)	26
3.3.4 Rancangan Proses Pemindahan Komponen Drum	28
3.3.5 Rancangan Perhitungan <i>Task Complete</i> pada Aplikasi Drum Virtual.....	28
3.3.6 Rancangan Perhitungan <i>Task Accuracy</i> pada Aplikasi Drum Virtual.....	29
3.3.7 Rancangan Perhitungan <i>Task Error</i> pada Aplikasi Drum Virtual.....	30
3.4 Perancangan Perangkat Lunak	30
3.4.1 Deskripsi Umum Perangkat Lunak.....	30
3.4.2 Spesifikasi Kebutuhan Fungsional.....	35
3.4.3 Spesifikasi Kebutuhan Non-Fungsional	35
3.4.4 Karakteristik Pengguna	36
3.5 Perancangan Sistem	36
3.5.1 Perancangan Diagram Kasus Penggunaan.....	36
3.5.2 Perancangan Skenario Kasus Penggunaan.....	38
BAB IV IMPLEMENTASI	41
4.1 Lingkungan Implementasi.....	41
4.2 Implementasi Aplikasi Drum Virtual	41
4.2.1 Implementasi Pembuatan <i>Scene</i>	42
4.2.2 Implementasi Pembuatan <i>Script</i>	44
4.2.3 Implementasi Tampilan Drum	45
4.2.4 Implementasi Pendeteksi Tumbukan objek drum dengan Stik.....	47
4.2.5 Implementasi Pemutaran Suara Drum	52
4.3 Implementasi Pemindahan Objek Drum yang Dapat Diatur oleh Pengguna	54
4.4 Implementasi Pendeteksi Jari.....	56
4.5 Implementasi Penerjemah Pola Tangan atau <i>Gesture</i>	66
4.6 Implementasi Pembuatan Kamera pada Unity	69
4.7 Implementasi Menampilkan Text pada Scene	70
4.8 Implementasi Penghitungan <i>Task Accuracy</i>	71

4.9 Implementasi Penghitungan Task Complete	73
4.10 Implementasi Penghitungan Task Error	74
4.11 Pembuatan Proyek	74
BAB V PENGUJIAN DAN EVALUASI	77
5.1 Lingkungan Uji Coba	77
5.1.1 Lingkungan Perangkat Keras	77
5.1.2 Lingkungan Perangkat Lunak	77
5.2 Pengujian Fungsionalitas	78
5.2.1 Cara Menjalankan Aplikasi	78
5.2.2 Skenario Uji Coba Fungsionalitas	78
5.2.3 Skenario Uji Coba Desain UX (<i>User Experience</i>)	79
5.2.4 Hasil Uji Coba	80
5.3 Pengujian Non-Fungsionalitas	86
5.3.1 Skenario Uji Coba Non-Fungsionalitas	86
5.3.2 Hasil Uji Coba	87
5.4 Hasil Uji Coba Dependent Variable	91
5.4.1 Hasil Uji Coba <i>Task Complete</i> Penggunaan Aplikasi	91
5.4.2 Hasil Uji Coba <i>Task Accuracy</i> Penggunaan Aplikasi	92
5.4.3 Hasil Uji Coba Jumlah <i>Task Error</i>	93
5.5 Evaluasi	94
5.5.1 Evaluasi Pengujian Fungsionalitas	94
5.5.2 Evaluasi Pengujian Non-Fungsionalitas	95
5.5.3 Evaluasi Pengujian Dependent Variable	97
BAB VI KESIMPULAN DAN SARAN	99
6.1. Kesimpulan	99
6.2. Saran	99
DAFTAR PUSTAKA	100
LAMPIRAN A KUISIONER PENGGUNA	103
LAMPIRAN B UJI COBA DEPENDET VARIABEL ...	109
BIODATA PENULIS	116

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 Perangkat Intel Realsense F200	7
Gambar 2.2 Komponen pada Perangkat Intel Realsense	8
Gambar 2.3 Cara Kerja Sensor IR	9
Gambar 2.4 Pendeteksi Pencerahan Objek pada Sensor IR	9
Gambar 2.5 Komponen Resonant Micromirror pada Perangkat Intel Realsense	10
Gambar 2.6 Bagian dari <i>User Interface</i>	13
Gambar 2.7 Bagian dari <i>User Experience</i>	14
Gambar 2.8 Berbagai Macam Alat Musik	15
Gambar 2.9 Alat Musik Drum.....	16
Gambar 2.10 Nama, Bentuk Not dan Tanda Istirahat serta Nilainya	19
Gambar 2.11 Nada pada Drum	19
Gambar 2.12 Tampilan Antarmuka Unity	20
Gambar 3.1 Penamaan Titik-Titik Ruas Jari	25
Gambar 3.2 Proses Pendeteksi Sendi pada Jari Tangan.....	25
Gambar 3.3 Proses Pemutaran Suara Saat Tangan Menyentuh Drum.....	26
Gambar 3.4 Tipe-Tipe Gesture yang dapat Dideteksi pada Perangkat Intel Realsense	27
Gambar 3.5 Proses Pendeteksi Gesture	27
Gambar 3.6 Proses Pemindahan Komponen Drum.....	28
Gambar 3.7 Proses Perhitungan <i>Task Complete</i>	29
Gambar 3.8 Simulasi Proses Perhitungan Task Accuracy	29
Gambar 3.9 Desain Aplikasi Tipe A: Aplikasi Air Drumming...	31
Gambar 3.10 Desain Aplikasi Tipe B	32
Gambar 3.11 Desain Aplikasi Tipe C	32
Gambar 3.12 Diagram kasus penggunaan.....	37
Gambar 4.1 Penambahan <i>Scene</i>	42
Gambar 4.2 Tampilan <i>Scene</i> Baru.....	43
Gambar 4.3 Tampilan <i>Project Explorer</i>	43
Gambar 4.4 Pembuatan <i>script</i> (1)	44
Gambar 4.5 Pembuatan <i>script</i> (2)	44

Gambar 4.6 Pembuatan <i>script</i> (3)	45
Gambar 4.7 Folder <i>Assets</i>	46
Gambar 4.8 Cara Membuat Folder Baru	46
Gambar 4.9 Cara Import <i>Asset</i> Baru	47
Gambar 4.10 Penambahan Komponen.....	50
Gambar 4.11 Pemilihan Mesh Collider.....	51
Gambar 4.12 Centang Is Trigger pada Mesh Collider	51
Gambar 4.13 Penambahan Sphere Collider pada Ujung Palm ...	52
Gambar 4.14 Pemilihan Menu Audio	52
Gambar 4.15 Pemilihan Menu Audio Source	53
Gambar 4.16 <i>Toggle Play On Awake</i> jangan dicentang.....	53
Gambar 4.17 <i>Inspector Transform</i>	54
Gambar 4.18 Cara Pembuatan Kamera	69
Gambar 4.19 Pembuatan <i>text</i> pada Unity.....	70
Gambar 4.20 Inspector <i>text</i> pada Unity.....	71
Gambar 4.21 Cara Membuat Proyek.....	75
Gambar 5.1 <i>Not-Not</i> Pada Task Pertama	80
Gambar 5.2 <i>Not-Not</i> pada Task Kedua	81
Gambar 5.3 <i>Not-Not</i> pada Task Ketiga	81
Gambar 5.4 <i>Not-Not</i> pada Task Keempat	81

DAFTAR TABEL

Tabel 2.1 Bagian-Bagian dari Komponen Drum	16
Tabel 3.1 <i>Independent Variable (IV)</i> dari Ketiga Desain Aplikasi	33
Tabel 3.2 Kuesioner Penggunaan Aplikasi	33
Tabel 3.3 <i>Dependent Variable</i> pada Aplikasi	34
Tabel 3.4 Experiment Task	35
Tabel 3.5 Karakteristik pengguna	36
Tabel 3.6 Skenario kasus penggunaan	37
Tabel 3.7 Skenario Kasus Penggunaan Memainkan Drum Virtual dengan Tangan	38
Tabel 3.8 Skenario Kasus Mengatur Posisi Alat Drum	39
Tabel 3.9 Skenario Kasus Mengatur Volume Suara	39
Tabel 4.1 Lingkungan implementasi perangkat lunak	41
Tabel 4.2 Penjelasan Kode Sumber 4-1	48
Tabel 4.3 Penjelasan Kode Sumber 4-2	49
Tabel 4.4 Penjelasan Kode Sumber 4-3	49
Tabel 4.5 Penjelasan Kode Sumber 4-4	55
Tabel 4.6 Penjelasan Kode Sumber 4-5	56
Tabel 4.7 Penjelasan Kode Sumber 4-6	58
Tabel 4.8 Penjelasan Kode Sumber 4-7	60
Tabel 4.9 Penjelasan Kode Sumber 4-8	62
Tabel 4.10 Penjelasan Kode Sumber 4-9	64
Tabel 4.11 Penjelasan Kode Sumber 4-10	65
Tabel 5.1 Lingkungan Perangkat Keras	77
Tabel 5.2 Lingkungan Perangkat Lunak	78
Tabel 5.3 Experiment Task	79
Tabel 5.4 Hasil Uji Coba Memainkan Drum Virtual	82
Tabel 5.5 Hasil Uji Coba Pemindahan Alat Musik Drum.....	84
Tabel 5.6 Uji Coba Pengaturan Volume Suara	85
Tabel 5.7 Daftar nama penguji coba aplikasi	87
Tabel 5.8 Rata-Rata Hasil Uji Kemudahan Penggunaan Aplikasi	88

Tabel 5.9 Detail Penilaian Kemudahan Penggunaan Apliaski Setiap Desain.....	88
Tabel 5.10 Rata-rata Hasil Uji Coba Kenaturalan Penggunaan Aplikasi	89
Tabel 5.11 Detail Penilaian Kenaturalan Penggunaan Aplikasi Setiap Desain.....	89
Tabel 5.12 Rata-Raja Hasil Uji Coba Kenyamanan Pengoperasian Aplikasi Setiap Desain	90
Tabel 5.13 Hasil Uji Coba Kenyamanan Pengoperasian Aplikasi Setiap Desain.....	90
Tabel 5.14 Rata-Rata Hasil Uji Coba Task Complete	91
Tabel 5.19 Rata-Rata Hasil Uji Coba <i>Task Accuracy</i>	92
Tabel 5.24 Total Hasil Uji Coba <i>Task Error</i> Penggunaan Aplikasi	93
Tabel 5.29 Rekapitulasi hasil uji coba fungsionalitas	95
Tabel 5.30 Rekapitulasi hasil uji coba non fungsionalitas	95
Tabel 5.31 Rekapitulasi hasil uji coba dependent variable	97

BAB I

PENDAHULUAN

1.1 Latar Belakang

Musik adalah suara yang disusun demikian rupa sehingga mengandung irama, lagu, dan keharmonisan terutama suara yang dihasilkan dari alat-alat yang dapat menghasilkan bunyi-bunyian. Walaupun musik adalah sejenis fenomena intuisi, untuk mencipta, memperbaiki dan mempersembahkannya adalah suatu bentuk seni [1]. Terdapat beberapa jenis alat musik berdasarkan cara memainkannya salah satunya adalah alat musik pukul yang menghasilkan suara sewaktu dipukul atau ditabuh. Alat musik pukul dibagi menjadi dua yakni bernada dan tidak bernada. Bentuk dan bahan bagian-bagian instrumen serta bentuk rongga getar, jika ada, akan menentukan suara yang dihasilkan instrumen. Contohnya adalah kolintang (bernada), drum (tak bernada), dan bongo (tak bernada) [2].

Terdapat alat musik dengan harga yang sulit untuk dijangkau oleh masyarakat luas salah satunya adalah alat musik drum. Harga alat musik ini sekitar 13 juta dalam satu setnya. Selain itu, dibutuhkan ruangan yang lebih luas untuk meletakkan alat musik ini. Hal ini mendasari pembuatan drum virtual seperti aplikasi drum virtual dengan bantuan Leap Motion pada aplikasi Air Drumming. Akan tetapi, aplikasi tersebut terdapat beberapa kelemahan seperti masih kurang interaktif untuk digunakan.

Intel Realsense adalah pengembangan teknologi dari *HCI* (*Human-Computer Interaction*) dimana di dalam *HCI* itu sendiri terdapat berbagai cabangnya diantaranya *UI* (*User Interface*) dan *UX* (*User Experience*). Intel Realsense menempatkan diri di teknologi tersebut sama seperti kompetitor lainnya yang menjadi teknologi awalnya yakni Kinect oleh Primesense, dan Leap Motion oleh Leap. Fitur yang dimiliki oleh Intel Realsense sendiri adalah dapat melakukan *facial analysis* seperti deteksi wajah, mata, ekspresi, *finger tracking* dimana kita bisa mendeteksi setiap sendi yang ada pada jari kita, *speech recognition* dimana kita bisa mengenali suara

dan bisa mengendalikan segala sesuatu di komputer dengan suara, *background subtraction*, dan *augmented reality*.

Tugas akhir ini mencoba untuk menerapkan desain *UX (User Experience)* pada aplikasi alat musik drum virtual menggunakan teknologi Intel Realsense. Teknologi Intel Realsense ini, memungkinkan pengguna dapat membunyikan suara dengan cara menggerakkan tangan seperti memukul drum pada kamera Intel Realsense. Tugas akhir ini diharapkan dapat memberikan pengalaman berbeda bagi pengguna saat memainkan aplikasi drum virtual dengan cara menggerakkan tangan.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana cara mengimplementasikan teknologi *finger tracking* Intel Realsense untuk membuat aplikasi drum virtual?
2. Bagaimana rekomendasi desain *UX (User Experience)* yang sesuai pada aplikasi drum virtual ini?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Aplikasi drum virtual ini dalam pengoperasiannya hanya menggunakan tangan dan tidak menggunakan kaki.
2. Aplikasi hanya dapat dijalankan pada perangkat *desktop*.

1.4 Tujuan Pembuatan Tugas Akhir

Tujuan dari pembuatan tugas akhir ini antara lain:

1. Membuat aplikasi drum virtual dengan bantuan teknologi Intel Realsense.
2. Membuat aplikasi drum virtual berbasis *desktop*.
3. Menerapkan desain *UX* yang sesuai pada aplikasi drum virtual.

1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini antara lain:

1. Dapat menjadi referensi bagi penelitian Tugas Akhir yang lain.
2. Menjelaskan mengenai prinsip-prinsip desain *UX* yang baik dan benar saat membangun suatu aplikasi.
3. Mengenalkan teknologi Intel Realsense salah satunya penggunaan fitur hand gesture dan finger tracking kepada masyarakat luas.

1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

A. Studi literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan tugas akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi, dan pustaka, serta pemahaman dasar teori yang digunakan pada topik tugas akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. Intel Realsense.
2. Alat musik drum.
3. Desain *UI (User Interface)*.
4. Desain *UX (User Experience)*.
5. SDK Intel Realsense.
6. Musik.
7. Unity.
8. Corel Draw.
9. Adobe Photoshop.

B. Perancangan perangkat lunak

Pada tahap ini diawali dengan melakukan analisis awal terhadap permasalahan utama yang muncul pada topik tugas akhir. Kemudian dilakukan perancangan perangkat lunak yang meliputi penentuan data yang akan digunakan dan proses-proses yang akan dilaksanakan. Langkah yang akan digunakan pada tahap ini adalah sebagai berikut:

1. Perancangan pendeteksi sendi pada jari (*finger tracking*).
2. Perancangan 3 desain *UX* aplikasi.
3. Perancangan fitur-fitur tambahan seperti pengaturan dasar aplikasi dan penyesuaian tata letak.

C. Implementasi dan pembuatan sistem

Pada tahap ini dilakukan membuat gambar objek-objek pada drum dengan menggunakan aplikasi Corel Draw dan Adobe Photoshop. Aplikasi ini dibangun menggunakan aplikasi Unity dengan bahasa dasar C# dan Javascript. Agar aplikasi ini dapat mendeteksi dengan perangkat Intel Realsense maka terlebih dahulu *PC (Personal Computer)* harus ter-*install* driver Intel Realsense yang dapat diunduh pada website resmi Intel Realsense.

D. Uji coba dan evaluasi

Pada tahap ini dilakukan uji coba dengan menggunakan beberapa macam kondisi untuk aplikasi bisa berjalan atau tidak. Uji fungsionalitas untuk mengetahui apakah aplikasi sudah memenuhi semua kebutuhan fungsional.

Pengujian aplikasi atau sistem dilakukan untuk mengukur suatu keberhasilan aplikasi ini dengan meminimalisir persepsi dan kesalahan-kesalahan. Pengujian aplikasi dengan cara menggunakan metode kuisisioner. Kuisisioner ini akan dibagikan kepada pengguna setelah mengoperasikan aplikasi drum virtual ini dan membandingkan hasil antara desain satu dengan desain yang lainnya. Selain itu untuk menganalisis kesalahan-kesalahan pengguna saat menggunakan aplikasi virtual drum, yang harus dilakukan adalah merekam layar aplikasi kemudian dicari kesalahan-kesalahan yang terjadi. Apabila hasil pengujian tersebut kurang memuaskan, maka yang harus dilakukan adalah memperbaiki aplikasi baik dari sisi desain ataupun program.

E. Penyusunan laporan tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Analisis dan Perancangan
4. Implementasi
5. Pengujian dan Evaluasi
6. Kesimpulan dan Saran

1.7 Sistematika Penulisan

Buku tugas akhir ini terdiri dari beberapa bab yang dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan tugas akhir, metodologi yang dapat digunakan, dan sistematika penyusunan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari dalam pembuatan tugas akhir ini.

BAB III

ANALISIS DAN PERANCANGAN

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, arsitektur, data, dan antarmuka.

BAB IV

IMPLEMENTASI

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembuatan drum virtual berbasis teknologi Intel Realsense.

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

BAB VI

KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Intel Realsense

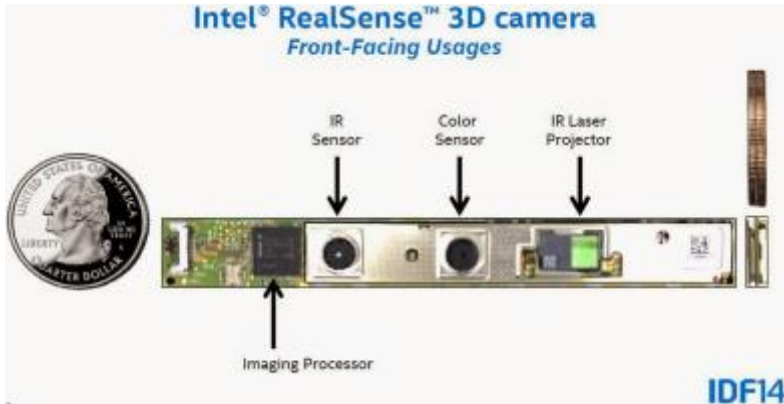
Intel Realsense adalah teknologi kamera yang mampu merespon tangan, lengan, dan gerakan kepala serta ekspresi wajah. kemampuan kamera Intel Realsense ini dapat mendeteksi 22 sendi di satu telapak tangan kita dan juga bisa mendeteksi kedalaman 3 dimensi. Berikut adalah perangkat Intel Realsense pada Gambar 2.1.



Gambar 2.1 Perangkat Intel Realsense F200 [3]

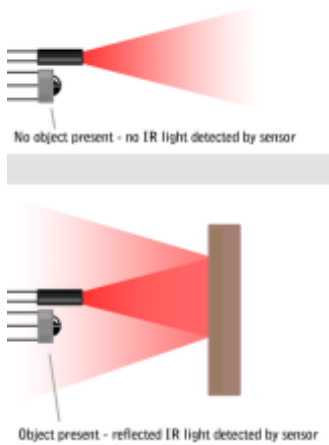
Rahasia dari kecanggihan teknologi Intel Realsense adalah perangkat dengan kamera Intel RealSense 3D memiliki tiga lensa. Jadi terdapat 3 kamera yang saling terintegrasi, tiga kamera tersebut adalah kamera konvensional, kamera inframerah, dan laser proyektor inframerah. Ketiga lensa kamera tersebut memungkinkan perangkat

untuk menyimpulkan kedalaman dengan mendeteksi cahaya inframerah dari benda-benda yang ada di depannya. Kemudian data visual yang diambil dikombinasi dengan *software* pelacak gerak Intel RealSense.

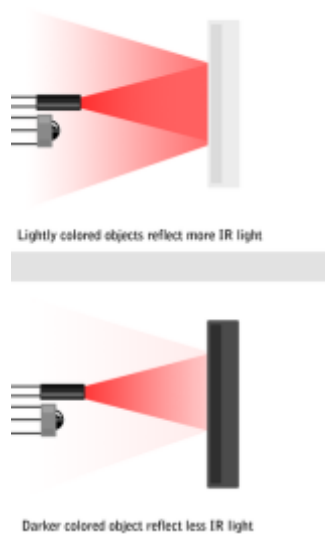


Gambar 2.2 Komponen pada Perangkat Intel Realsense [4]

Intel Realsense memiliki 3 sensor dengan fungsi yang berbeda seperti terlihat pada Gambar 2.2. Ketiga sensor tersebut akan meneruskan data yang sudah diperoleh ke komponen *Imaging Processor*. Sensor yang pertama bernama *IR (Infra-Red) Sensor* atau sensor inframerah yang memiliki fungsi untuk menghitung itentitas cahaya (*brightness*). Prinsip kerjanya yaitu gelombang inframerah dipancarkan kedepan kamera, kemudian gelombang akan memantul balik ketika terkena dengan benda seperti pada Gambar 2.3. Itensitas cahaya yang dipantulkan akan semakin besar apabila terkena benda yang memiliki itensitas cahaya lebih terang. *Infra-Red Sensor* ini biasanya digunakan pada kamera digital dan robot *Line Tracker*. Untuk lebih jelasnya dapat dilihat pada Gambar 2.4.



Gambar 2.3 Cara Kerja Sensor IR [4]



Gambar 2.4 Pendeteksi Pencerahan Objek pada Sensor IR [4]

Sensor yang kedua adalah *Color Sensor* yang berfungsi untuk menerima warna pada gambar. Sensor ini dapat menerima warna dengan format *Red Green Blue (RGB)* pada permukaan objek. Kemudian warna yang sudah diperoleh akan dikonversikan ke dalam bentuk *pixel*. Sensor warna ini biasanya digunakan pada perangkat kamera digital. *Intel Realsense* ini dapat mengambil gambar dengan ukuran maksimal *1080p*.

Sensor yang ketiga adalah *IR Laser Projector*. Sensor ini berguna untuk mendeteksi kedalaman atau jarak terhadap objek. Sensor ini dapat menangkap objek 3 dimensi sehingga alat ini dapat digunakan untuk memindai objek dan kemudian dikonversi dalam gambar 3 dimensi. Prinsip kerjanya adalah sensor ini akan memancarkan resonansi ke depan kamera dan apabila resonansi tersebut bertabrakan dengan objek, maka resonansi akan memantul kembali. Dengan begitu sensor ini akan menerima jarak/kedalaman suatu objek. Resonan pada *IR Laser Projector* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Komponen Resonant Micromirror pada Perangkat Intel Realsense [5]

2.2 Desain UI (*User Interface*)

User Interface adalah cara program dan *user* berkomunikasi. Istilah *User Interface* atau *Interface* kadang-kadang digunakan sebagai pengganti istilah *HCI* (*Human Computer Interaction*). *HCI* (*Human Computer Interface*) adalah semua aspek dari interaksi pengguna dan komputer, tidak hanya *hardware*. Semuanya yang terlihat di layar, membaca dalam dokumentasi dan dimanipulasi dengan *keyboard* (atau *mouse*) merupakan bagian dari *User Interface*.

User Interface berfungsi untuk menghubungkan atau penterjemah informasi antara pengguna dengan sistem operasi, sehingga komputer dapat digunakan. Dengan demikian, *User Interface* bisa juga diartikan sebagai mekanisme inter-relasi atau integrasi total dari perangkat keras dan lunak yang membentuk pengalaman bekomputer. *User Interface* dari sisi *software* bisa berbentuk *Graphical User Interface* (*GUI*) atau *Command Line Interface* (*CLI*), sedangkan dari sisi *hardware* bisa berbentuk *Apple Desktop Bus* (*ADB*), *USB*, dan *fire wire*.

Konsep *User Interface* secara benar tidaklah mudah. Terdapat begitu banyak aspek yang perlu diperhatikan. *User Interface* akan mengacu pada beragam aplikasi teknologi mulai dari *electronic display*, *software* aplikasi komputer, aplikasi *website*, aplikasi *mobile*, hingga aplikasi *kioks* Informasi public. *Kioks* adalah peralatan sistem informasi publik yang dirancang sedemikian rupa yang ditujukan untuk beragam kondisi user, baik secara usia, gender, latar belakang kultural, tingkat pemahaman dan pendidikan bahkan kondisi keterbatasan fisik yang berbeda.

Terdapat dua jenis *User Interface* yaitu:

1. *Command Line Interface* (*CLI*)

CLI (*Command Line Interface*) adalah tipe antarmuka dimana pengguna berinteraksi dengan sistem operasi melalui *text terminal*. *CLI* adalah sebuah bentuk antarmuka antara sistem operasi dan pemakai dimana pemakai mengetikkan perintah-perintah dengan menggunakan perintah dalam bentuk teks dan sebuah metode untuk memasukinya.

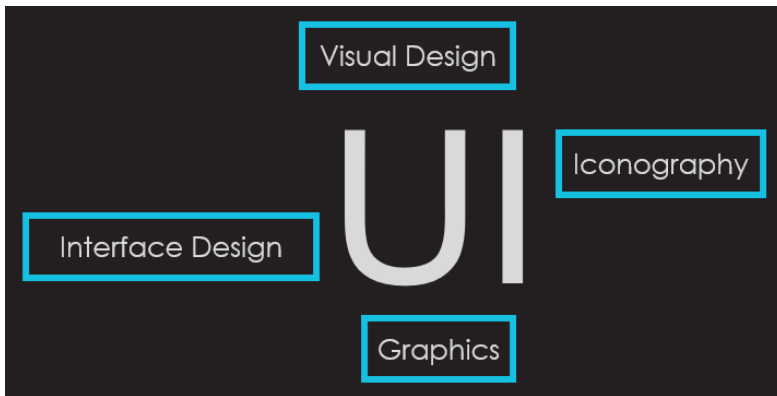
Pengguna *CLI* biasanya adalah administrator sistem berbasis sistem operasi *LINUX*. Setiap sistem operasi memberi nama *CLI*-nya berbeda-beda. *Unix* member nama *CLI*-nya sebagai *bash*, *ash*, *ksh*, dan lain sebagainya. *Ms-Dos* memberi nama *CLI*-nya *command.com* atau *command prompt*. Sedangkan *Windows Vista*, *Microsoft* menamakannya *Powershell*. Pengguna *Linux* mengenal *CLI* pada *Linux* sebagai *Terminal*, sedangkan pada *Apple* atau *machintosh* namanya adalah *commandshell*.

2. *Graphical User Interface (GUI)*

Saat ini *Interface* yang banyak digunakan dalam *software* adalah *GUI (Graphical User Interface)*. Penganut *GUI* biasanya adalah mereka yang sudah terbiasa dengan sistem operasi *Windows*. Bagi mereka, *GUI* adalah harga mati yang tidak bisa ditawar lagi.

GUI adalah tipe antarmuka yang digunakan oleh pengguna untuk berinteraksi dengan sistem operasi melalui gambar-gambar grafik, ikon, dan menggunakan perangkat penunjuk (*pointing device*) seperti *mouse* atau *track ball*.

Sama seperti *CLI*, tiap-tiap sistem operasi memiliki nama tersendiri untuk komponen *GUI*-nya. Pada *Apple Mac OS X*, *GUI*-nya disebut *Aqua*. *Microsoft* member nama *GUI* pada *Windows XP* sebagai *Lunar* dan *GUI Windows Vista* sebagai *Aero*. Pada *Linux*, ada dua pengembangan utama *desktop environment*, yang masing-masing menghasilkan produk *KDE (K Desktop Environment)* dan *GNOME*.



Gambar 2.6 Bagian dari *User Interface* [6]

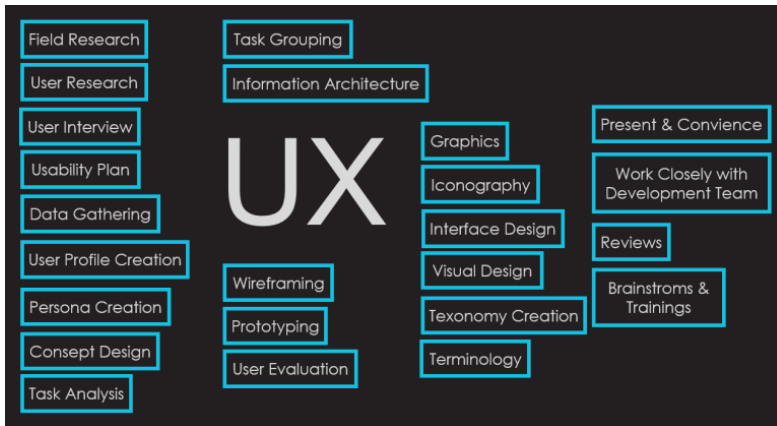
Pada Gambar 2.6 terlihat bahwa *User Interface* merupakan gabungan dari beberapa bagian seperti *Visual Design*, *Iconography*, *Interface Design* dan *Graphics*.

2.3 Desain UX (*User Experience*)

User Experience memiliki ranah yang lebih luas dari *UI*, karena ranah *UX* ini dimulai dengan *research* pasar yang kemudian diimplementasi ke dalam sebuah *interface*. *UI* adalah bagian dari *UX* dimana *UI* merupakan produk akhir dari *UX*. User experience designer mampu menghasilkan *User Interface* yang mudah digunakan oleh target pengguna. Karena ranah *UX* ini terlihat cukup luas, pada beberapa perusahaan memecah *UX designer* menjadi 2 kelompok, *UX Researcher* dan *UI Designer* untuk menciptakan *User Experience* dari suatu *apps* atau *website* yang baik.

Teknologi digital tidak sama dengan teknologi lainnya. Misalnya, kursi adalah mesin kompleks yang menyediakan layanan permintaan pengguna, namun orang secara intuitif tahu bagaimana menggunakan kursi. Kita memahami tujuan dan perilaku kursi karena kita telah berevolusi dengan kursi tersebut. Tidak seperti halnya dengan kursi, manusia tidak berevolusi menggunakan teknologi digital, dan karena itu kita tidak secara naluriah memahami cara kerjanya.

Dalam desain *UX* terdapat 3 pertanyaan dalam memvalidasi suatu aplikasi yaitu: Siapa penggunanya?, Bagaimana mereka menggunakannya?, dan apakah sistem tersebut dapat bekerja?. Kemudian dari validasi akan membantu menciptakan suatu desain aplikasi yang nantinya aplikasi tersebut dapat mencapai sasarnya [7].



Gambar 2.7 Bagian dari *User Experience* [6]

UX lebih kompleks dibandingkan dengan *UI* selain itu *UI* merupakan bagian dari *UX* terlihat terdapat bagian *UI* seperti *Visual Design*, *Iconography*, *Interface Design* dan *Graphics* terlihat pada Gambar 2.7.

2.4 Musik

Musik adalah curahan hati melalui bunyi sebagai perantaranya. Maksudnya bahwa musik adalah salah satu cabang seni abstrak yang berbentuk suara dan terdiri atas unsur-unsur ritme, melodi, harmoni, serta timbre. Musik adalah ilmu atau seni menyusun nada atau suara diurutkan, kombinasi, dan hubungan temporal untuk menghasilkan komposisi (suara) yang mempunyai kesatuan dan kesinambungan. Berikut adalah berbagai macam alat musik terlihat pada Gambar 2.8.



Gambar 2.8 Berbagai Macam Alat Musik

Unsur-unsur musik terdiri dari beberapa kelompok yang secara bersama merupakan satu kesatuan membentuk suatu lagu atau komposisi musik. Semua unsur musik tersebut berkaitan erat dan sama-sama mempunyai peranan penting dalam sebuah lagu.

Pada dasarnya unsur-unsur musik dapat dikelompokkan unsur pokok yaitu harmoni, irama, melodi, atau struktur lagu dan unsur-unsur ekspresi yaitu tempo, dinamika dan warna nada, kedua unsur musik tersebut merupakan satu kesatuan yang tidak dapat dipisahkan.

2.5 Alat Musik Drum


Drum adalah kelompok alat musik perkusi yang terdiri dari kulit yang direntangkan dan dipukul dengan tangan atau sebuah batang. Selain kulit, drum juga digunakan dari bahan lain, misalnya plastik. Dalam musik *pop*, *rock*, dan *jazz*, drums biasanya mengacu kepada *drum kit* atau drum set, yaitu sekelompok drum yang biasanya terdiri dari *snare drum*, *tom-tom*, *bass drum*, *cymbal*, *hi-hat*, dan kadang ditambah berbagai alat musik drum listrik. Orang yang memainkan drum set disebut “drummer”. Berikut adalah contoh alat musik drum terlihat pada Gambar 2.9.



Gambar 2.9 Alat Musik Drum

Berikut ini adalah tabel komponen dari drum set terlihat pada Tabel 2.1.

Tabel 2.1 Bagian-Bagian dari Komponen Drum

No	Komponen Drum	Nama Komponen Drum
1		<i>Bass Drum</i>





2		<i>Snare Drum</i>
3		<i>Tom Tom</i>
4		<i>Cymbol</i>

5		<i>Hi hat</i>
---	---	---------------

Biasanya drum dibagi berdasarkan kelasnya, ini dibagi menurut tingkat harganya. Ada yang disebut entry level atau drum untuk pemula. Ada juga drum untuk kelas menengah dan ada juga drum untuk kelas atas. Biasanya drum untuk kelas menengah dan kelas atas, membeli drum tidak disertai dengan *cymbal*. Untuk cymbal harus membeli tersendiri.

2.6 Not Balok Drum

Seperti alat musik lain, not balok pada drum juga menggunakan garis penanda dan beberapa simbol-simbol standar. Tetapi pada not drum terdapat beberapa simbol not yang tidak terdapat pada not alat musik lain. Setiap simbol not dan tanda istirahat mempunyai nama dan nilai, seperti Not Penuh (*Whole Note*) yang nilainya 4 ketuk, Not 1/2 yang nilainya 2 ketuk dan seterusnya. Tetapi pada drum, Not Penuh dan Not 1/2 sangat jarang digunakan karena drum tidak memiliki sustain yang panjang. Nilai pada not dan tanda istirahat dapat dilihat pada gambar berikut Gambar 2.10.

Nama Not	Bentuk Not	Tanda Istirahat	Nilai
Not Penuh			4 Ketuk
Not 1/2			2 Ketuk
Not 1/4			1 Ketuk
Not 1/8			1/2 Ketuk
Not 1/16			1/4 Ketuk
Not 1/32			1/8 Ketuk

Gambar 2.10 Nama, Bentuk Not dan Tanda Istirahat serta Nilainya [8]

Sementara itu tangga nada pada alat musik drum beserta nama komponen yang dimainkan adalah pada Gambar 2.11.



Gambar 2.11 Nada pada Drum [8]

2.7 Unity

Aplikasi unity 3D adalah *game engine* merupakan sebuah *software* pengolah gambar, grafik, suara, input, dan lain-lain yang ditujukan untuk membuat suatu *game*, meskipun tidak selamanya harus untuk *game*. Contohnya adalah seperti materi pembelajaran untuk simulasi membuat SIM. Kelebihan dari *game engine* ini adalah bisa membuat *game* berbasis 3D maupun 2D, dan sangat mudah digunakan [9]. Berikut adalah aplikasi Unity pada Gambar 2.12.

Unity merupakan *game engine* yang ber-*multiplatform*. Unity mampu di publish menjadi *Standalone* (.exe), berbasis web, berbasis web, Android, iOS Iphone, XBOX, dan PS3. Dengan Unity3D kita dapat membuat *game* 3D, FPS dan 2D *game* bahkan *Game Online*.



Gambar 2.12 Tampilan Antarmuka Unity [10]

2.8 Diagram Use Case

Use Case adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. *Use case*

Diagram menampilkan aktor mana yang menggunakan *use case* mana, *use case* mana yang memasukkan *use case* lain dan hubungan antara aktor dan *use case* [8].

(Halaman ini sengaja dikosongkan)

BAB III

ANALISIS DAN PERANCANGAN

Bab ini menjelaskan tentang analisis dan perancangan aplikasi drum virtual berbasis teknologi Intel Realsense. Pembahasan yang akan dilakukan meliputi analisis fitur yang dibutuhkan dan perancangan perangkat lunak.

3.1 Analisis Sistem

Aplikasi ini dibangun sebagai alternatif alat musik drum. Karena alat musik ini memiliki harga yang tergolong mahal, sehingga orang masih kesulitan untuk memperolehnya. Selain itu alat musik ini membutuhkan ruangan yang lebih luas karena alat musik ini tergolong alat dengan ukuran besar. Drum mengeluarkan suara keras yang dapat mengganggu pendengaran. Untuk meredam suara keras tersebut maka di setiap dinding dibutuhkan peredam suara berupa gabus atau karpet. Dari beberapa kekurangan tersebut maka perlu dibutuhkan aplikasi drum virtual sebagai alternatif dari drum aslinya, sehingga pengguna tidak perlu membeli alat musik drum yang tergolong mahal tersebut.

Aplikasi drum virtual ini dibangun menggunakan teknologi Intel Realsense sehingga pengguna dapat memainkannya menggunakan gerakan tangan. Aplikasi ini dibangun menggunakan aplikasi Unity karena fitur-fitur pada aplikasi ini tergolong lengkap dan mudah untuk dipahami. Fitur tersebut seperti pembuatan kamera, suara, pendeteksi *collision*, gambar, objek 2 atau 3 Dimensi (3D) dan lainnya. Kemudian aplikasi drum virtual ini akan dilakukasn uji coba terhadap 3 desain dengan *UI (User Interface)* serta *UX (User Experience)* yang berbeda.

Dari ketiga desain yang telah dibuat pengguna diberi *task* kemudian akan dihitung nilai dari kuisioner yang telah diisi oleh pengguna serta akan diambil data. Kuisioner tersebut berupa nilai kemudahan penggunaan aplikasi, nilai kenaturalan aplikasi, nilai kenyamanan penggunaan aplikasi. Selain ketiga nilai tersebut pada saat pengguna melakukan *task* yang telah diberikan pada aplikasi akan dihitung berapa lama pengguna dapat menyelesaikan *task* tersebut,

seberapa akurasi penggunaan drum virtual dan berapa kesalahan pengguna.

3.2 Perancangan UI (*User Interface*) Alat Musik Drum

Perancangan UI (*User Interface*) aplikasi drum virtual dibuat menggunakan Corel Draw X5 dan Adobe Photoshop CS6. Pembuatan desain gambar meliputi komponen pada drum seperti *Snare*, *Bass Drum*, *tom-tom*, *cymbal*, *hithat*. Perancangan komponen tersebut menggunakan Corel Draw. Desain yang dibuat merupakan desain 2D. Setelah itu, untuk memberikan kesan lebih nyata, kemudian hasil perancangan dari aplikasi Corel Draw dimanipulasi menggunakan Adobe Photoshop seperti pewarnaan, pemberian efek bayangan, dan pembenahan warna.

3.3 Perancangan Proses

Pada subbab ini akan dijelaskan rancangan proses yang akan digunakan untuk memenuhi kebutuhan fungsionalitas perangkat lunak yang dibangun. Berikut ini adalah penjelasan dari rancangan algoritma yang digunakan dalam perangkat lunak ini.

3.3.1 Rancangan Proses Pendeteksi Sendi pada Jari Tangan

Aplikasi yang akan dibuat merupakan aplikasi virtual drum yang menggantikan alat musik drum aslinya. Aplikasi ini menggunakan gerakan kedua tangan sebagai input utama. Sehingga aplikasi ini harus dapat mendeteksi gerakan jari. Untuk itu dibutuhkan alat Intel Realsense yaitu sensor untuk menerima gerakan jari tangan. Kemudian agar data gerakan jari dapat digunakan pada aplikasi, maka diperlukan Intel Realsense SDK. terdapat beberapa fitur pada Intel Realsense SDK salah satunya adalah *finger tracking*.

Konsep dari *finger tracking* dalam kamera Intel Realsense akan menerima data kedalaman objek (tangan) di depan kamera. Jarak yang dapat dijangkau kamera ini adalah maksimal 25 cm. Data yang diinputkan pada kamera ini adalah data *Binary Large Object (BLOB)*. Kemudian *processor* Intel (minimum generasi 4 ke atas) akan memproses data *BLOB* tersebut dengan algoritma *Image Processing* sehingga akan menampilkan titik-titik sendi jari. Setiap titik pada

sendi ruas jari memiliki penamaan tersendiri yang dapat dilihat pada Gambar 3.1. Proses pendeteksi gerakan tangan menggunakan aplikasi Unity pada Gambar 3.2.



Gambar 3.1 Penamaan Titik-Titik Ruas Jari [11]

1. Inisialisasi *PXCMsenseManager* (Intel Realsens SDK).
2. Mengaktifkan *hand tracking* dan *hand module*.
3. Inisialisasi eksekusi *pipeline*.
4. Penerimaan *DataSmoothing instance*.
5. Pembuatan *3D Weighted algorithm*.
6. Mengaktifkan *gesture* dan modul lainnya.
7. Inisialisasi jumlah tangan dan jumlah ruas jari.
8. Tampilkan titik pembatas ruas jari dan penghubung ruas jari.
9. Ambil koordinat titik-titik pembatas ruas jari dalam bentuk koordinat x, y dan z.
10. Tiap titik-titik pembatas ruas jari ditampilkan pada layar.

Gambar 3.2 Proses Pendeteksi Sendi pada Jari Tangan

3.3.2 Rancangan Proses Pemutaran Suara Saat Tangan Menyentuh Drum

Ketika titik-titik ruas jari berhasil ditampilkan di layar dengan fitur *finger tracking* pada Intel Realsense, maka tahapan selanjutnya adalah memilih salah satu titik yang nantinya diubah menjadi alat stik drum. Titik tersebut adalah berupa titik *Palm* seperti pada Gambar 3.2. Pemilihan titik ini dikarenakan titik *Palm* lebih stabil pendeteksiannya dibandingkan titik lainnya. Ketika titik ini menyentuh drum maka akan menghasilkan suara drum seperti pada alat musik drum aslinya. Proses yang terdapat pada subbab ini yaitu melakukan pemutaran suara dalam format *.wav* (karena ekstensi ini yang dapat dimainkan pada Unity) yang berisi suara setiap komponen alat musik drum. Caranya adalah membuat sebuah objek (komponen alat musik drum) yang dapat disentuh, sehingga apabila objek tersebut tersentuh, maka sebuah media *.wav* tersebut akan diputar. Proses lebih lanjut akan dijelaskan pada Gambar 3.3.

1. Buat objek pada *Unity*, tambahkan komponen *Collider* pada objek tersebut. Kemudian tambahkan efek *physics Rigidbody*.
2. Tambahkan komponen *audio source* pada objek, lakukan pengaitan berkas *audio* terhadap komponen ini, lalu hilangkan centang *start on awake* terhadap komponen ini agar *audio* tidak terputar saat awal aplikasi dijalankan.
3. Berikan kode pemutaran dengan peristiwa titik *Palm* menyentuh objek.

Gambar 3.3 Proses Pemutaran Suara Saat Tangan Menyentuh Drum

3.3.3 Rancangan Proses Pendeteksi *Gesture* (Pola Tangan)

Intel Realsense SDK memiliki algoritma pendeteksi *gesture* atau pola tangan seperti *spreadfingers*, *first*, *tap*, *thumb_up*, *thumb_down* dan *gesture* lainnya yang dapat dilihat pada Gambar 3.4. *Gesture* ini dapat berguna sebagai pananda aksi seperti ketika pengguna melakukan pola *full_pinch* maka dapat diartikan sebagai *click* objek atau memilih objek. Pada aplikasi yang akan dibangun,

penggunaan *gesture* akan dimanfaatkan untuk navigasi, pemilihan objek dan pengaturan lainnya. Berikut adalah proses pendeteksi *gesture* pada aplikasi yang akan dibangun pada Gambar 3.5.



Gambar 3.4 Tipe-Tipe Gesture yang dapat Dideteksi pada Perangkat Intel Realsense [12]

1. Inisialisasi *PXCMSenseManager* (Intel Realsens SDK).
2. Mengaktifkan *hand tracking* dan *hand module*.
3. Inisialisasi eksekusi *pipeline*.
4. Penerimaan *DataSmoothing instance*.
5. Pembuatan *3D Weighted algorithm*.
6. Mengaktifkan *gesture* dan modul lainnya.
7. Inisialisasi jumlah tangan dan jumlah ruas jari.
8. Pemilihan tangan yang sudah terinisialisasi apakah tangan kiri atau kanan.
9. Analisis tipe *gesture* yang ditampilkan.
10. Jika *gesture* berupa *full_pinch* maka pilih objek.

Gambar 3.5 Proses Pendeteksi Gesture

3.3.4 Rancangan Proses Pemindahan Komponen Drum

Aplikasi yang akan dibangun akan memiliki fitur tambahan yaitu pengguna dapat memindahkan komponen-komponen alat musik drum sesuai kebutuhan dan kenyamanan pengguna. Hal ini dikarenakan setiap pengguna dengan pengguna lainnya memiliki panjang tangan dan kontraksi otot yang berbeda-beda. Demi kenyamanan pengguna saat memainkan alat musik drum maka fitur ini sangat diperlukan. Proses lebih lanjut akan dijelaskan pada gambar Gambar 3.6.

1. Buat objek baru seperti *cylinder*, kemudian tambahkan komponen *collider* dan centang *Trigger*.
2. Buat script baru kemudian tambahkan pada objek tersebut. Pada script buat fungsi *OnCollisionEnter*, *OnCollisionStay* dan *OnCollisionExit*.
3. Pada fungsi *OnCollisionEnter*, buat *code* untuk memindahkan objek.
4. Pada objek stik tambahkan komponen *physics collider* agar dapat mendeteksi tumbukan dengan komponen drum.

Gambar 3.6 Proses Pemindahan Komponen Drum

3.3.5 Rancangan Perhitungan *Task Complete* pada Aplikasi Drum Virtual

Cara menghitung *task complete* adalah dengan cara menghitung seberapa lama pengguna mengerjakan task yang sudah disediakan untuk dilakukan pengujian Berikut adalah proses perhitungan *task complete* pada aplikasi drum virtual pada .

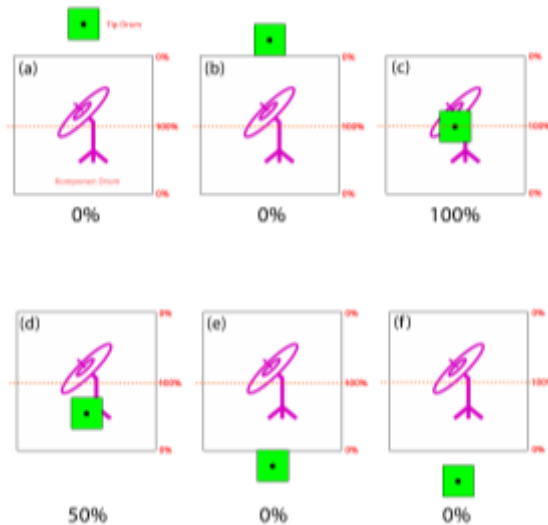
1. Buat variabel fungsi waktu dan dipanggil saat pertama kali aplikasi dijalankan.
2. Jika pengguna mengerjakan task dengan benar tanpa melakukan kesalahan (memukul drum lain) maka variabel untuk menampung jumlah *task* benar yang harus dilakukan bertambah.

3. Saat jumlah variabel melakukan *task* benar sama dengan jumlah maksimal maka waktu di *stop*.

Gambar 3.7 Proses Perhitungan *Task Complete*

3.3.6 Rancangan Perhitungan *Task Accuracy* pada Aplikasi Drum Virtual

Perhitungan *task accuracy* digunakan untuk mengukur seberapa akurasi pengguna mengerjakan *task* yang telah disediakan. Caranya adalah menghitung seberapa akurasi *tempo* pukulan drum pengguna dengan *task* yang telah tersedia.



Gambar 3.8 Simulasi Proses Perhitungan *Task Accuracy*

Pada Gambar 3.8 (a), (b), (e) dan (f) terlihat bahwa, saat *tip drum* tidak menyentuh komponen drum, maka jika pengguna memukul komponen drum tersebut, nilai akurasinya adalah 0 %. Kemudian pada (c) nilai akurasinya adalah 100% saat pengguna memukul komponen drum, karena *tip drum* berada tepat di tengah

komponen drum. Dan pada (d) nilai akurasinya adalah 50% karena *tip drum* berada di tengah nilai 0% dan 100%.

3.3.7 Rancangan Perhitungan *Task Error* pada Aplikasi Drum Virtual

Perhitungan *task error* digunakan untuk mengukur berapa kali pengguna melakukan kesalahan yaitu memukul komponen drum yang salah. Caranya adalah pada saat status tip drum berada pada Gambar 3.8 (a), (b), (e) dan (f) apabila saat itu pengguna memukul alat drum tersebut maka nilai kesalahannya bertambah.

3.4 Perancangan Perangkat Lunak

Pada subbab ini akan dibahas mengenai deskripsi umum perangkat lunak, spesifikasi kebutuhan fungsional dan kebutuhan non-fungsional serta bagaimana karakteristik pengguna aplikasi.

3.4.1 Deskripsi Umum Perangkat Lunak

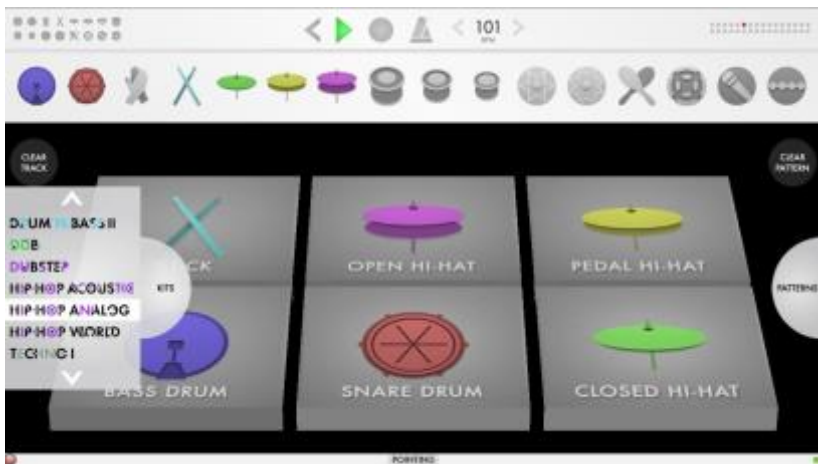
Aplikasi yang akan dibangun adalah sebuah aplikasi yang memvisualisasikan alat musik drum. Aplikasi virtual drum ini akan menampilkan perangkat drum (drumset) seperti *snare drum*, *bass drum*, *tom-tom*, *floor tom*, *hi-hat*, *ride cymbal*, dan *crash cymbal*. Sudut pandang yang digunakan pada aplikasi virtual drum ini adalah mengambil dari sisi atas. Sementara itu, suara yang dihasilkan oleh aplikasi akan disamakan dengan suara drum aslinya.

Cara memainkan aplikasi drum virtual ini menggunakan pergerakan tangan. Tidak seperti drum aslinya, cara memainkan bass drum pada aplikasi drum virtual ini juga menggunakan pergerakan tangan bukan menggunakan gerakan kaki. Pergerakan tangan yang dimaksud adalah tangan pengguna digerakan maju secara cepat menuju kamera Intel Realsense dengan menyesuaikan gambar perangkat drum pada layar *desktop*.

Pada saat perancangan yang harus diperhatikan adalah bagaimana aplikasi tersebut mudah untuk dioperasikan bagi pengguna. Karena cara mengoperasikan aplikasi ini menggunakan pergerakan tangan, maka perlu ada tutorial atau cara penggunaan aplikasi ini, yaitu dengan memberikan simulasi pada awal aplikasi

virtual drum ini. Selain itu, saat mendesain tampilan aplikasi ini, tata letak tampilan juga perlu diperhatikan dengan mengutamakan desain *UX (User Experience)* yang benar dan baik agar persepsi pengguna sama dengan persepsi yang seharusnya.

Terdapat 3 pilihan desain aplikasi yang akan disediakan pada aplikasi ini. Yang pertama yaitu tampilan aplikasi yang disesuaikan dengan aplikasi Air Drumming seperti pada Gambar 3.9. Tampilan yang kedua adalah tampilan yang dibuat oleh penulis pada Gambar 3.10. Tampilan yang ketiga adalah tampilan aplikasi yang mengadopsi dari aplikasi Drums seperti pada Gambar 3.11. Hal yang membedakan antara ketiga desain tersebut terletak pada Tabel 3.1.



Gambar 3.9 Desain Aplikasi Tipe A: Aplikasi Air Drumming [13]



Gambar 3.10 Desain Aplikasi Tipe B



Gambar 3.11 Desain Aplikasi Tipe C [14]

Tabel 3.1 *Independent Variable (IV)* dari Ketiga Desain Aplikasi

<i>Factor (IV)</i>	<i>Levels (Test Condition)</i>		
	Desain A	Desain B	Desain C
Dimensi visualisasi	3 Dimensi	2 Dimensi	3 Dimensi
Tata letak vnstrumen	Sebaris dan sejajar	Melingkar	Tata letak drum pada umumnya
Bentuk instrumen	Tidak natural (kotak dengan gambar instrumen)	Tidak natural (lingkaran dengan gambar instrumen)	Natural (berbentuk seperti instrumen aslinya)
Keterangan nama alat	Ada	Ada	Ada
Ukuran	Sama	Sama	Beragam
Jarak	Berdekatan	Ada berdekatan ada yang berjauhan	Berdekatan

Dari ketiga tampilan tersebut akan dibandingkan *UX* mana yang lebih baik. Parameter yang dinilai adalah dengan membandingkan hasil dari kuesioner yang diisi oleh pengguna seperti pada Tabel 3.2. Pertanyaan yang akan diajukan seperti apakah aplikasi dengan desain tipe A, B atau C mudah digunakan, apakah aplikasi dengan desain tipe A, B atau C lebih natural dan nyaman dengan lainnya [15].

Tabel 3.2 Kuesioner Penggunaan Aplikasi

	<i>Key Feature</i>	Pertanyaan
1	Kemudahan penggunaan	Berapa rating kemudahan penggunaan aplikasi dengan desain tipe (x)?
2	Kenaturalan	Berapa rating kenaturalan penggunaan aplikasi dengan desain tipe (x)?

3	Kenyamanan	Berapa rating kenyamanan penggunaan aplikasi dengan desain tipe (x)?
---	------------	--

Selain itu, pada ketiga aplikasi tersebut akan diukur *Dependent Variable*. *Dependent Variable* adalah suatu ukuran untuk perilaku manusia (berhubungan dengan aspek interaksi yang melibatkan *Independent Variable*). Dikatakan *Dependent* karena tergantung pada apa yang dilakukan oleh partisipan. *Dependent variables* harus didefinisikan secara jelas. Contoh dari *Dependent Variable* adalah *task completion time*, *speed*, *accuracy* dan *error rate*. Pada kasus aplikasi ini terdapat beberapa *Dependent Variable* yang akan didapat seperti pada Tabel 3.3.

Tabel 3.3 *Dependent Variable* pada Aplikasi

	<i>Dependent Variable</i>	Keterangan
1	<i>Task completion time</i>	Pegguna memainkan aplikasi drum virtual sesuai naskah <i>drum beats</i> yang diberikan. Kemudian dihitung waktu penyelesaian pekerjaan tersebut. Waktu penyelesaian harus sesuai dengan tempo pada naskah <i>drum beats</i> .
2	<i>Accuracy</i>	Saat pengguna memainkan aplikasi drum virtual akan dihitung tingka akurasi tempo pemukulan alat drum virtual tersebut.
3	<i>Error rate</i>	Saat pengguna memainkan aplikasi drum akan dilakukan seberapa sering pengguna melakukan kesalahan seperti salah memukul drum.

Selanjutnya, agar hasil dari kuisioner menjadi valid karena terdapat 3 desain aplikasi maka diperlukan *Experiment Task*. *Experiment Task* adalah *task* yang harus diselesaikan oleh partisipan sehingga pengaruh dari *independent variabel* terhadap perilaku partisipan terlihat. *Experiment task* yang bagus adalah yang *representative* dan *discriminative*. *Representative* artinya merepresentasikan aktivitas partisipan dengan *interface*.

Discriminative artinya mampu membedakan antara *test condition* yang berbeda-beda. *Experiment Task* dilakukan menggunakan Latin Square 3 x 3. Terdapat 6 pengguna kemudian pengguna tersebut dibagi menjadi 3 kelompok (I, II, III), masing-masing kelompok 2 orang. Kemudian kelompok tersebut memainkan urutan desain aplikasi drum virtual sesuai baris pada Tabel 3.4.

Tabel 3.4 Experiment Task

Kelompok	Uji Coba Ke-1	Uji Coba Ke-2	Uji Coba Ke-3
Kelompok I	A	B	C
Kelompok II	B	C	A
Kelompok III	C	B	A

3.4.2 Spesifikasi Kebutuhan Fungsional

Berdasarkan deskripsi umum sistem, maka disimpulkan bahwa kebutuhan fungsional dari aplikasi ini yaitu dapat memainkan aplikasi drum virtual pada setiap desain dan dapat mengubah letak dan ukuran drum virtual sesuai keinginan pengguna.

3.4.3 Spesifikasi Kebutuhan Non-Fungsional

Terdapat beberapa kebutuhan non-fungsional yang apabila dipenuhi, dapat meningkatkan kualitas dari aplikasi ini. Berikut daftar kebutuhan non-fungsional:

1. Kemudahan Penggunaan Aplikasi

Aplikasi yang baik adalah aplikasi yang mudah untuk digunakan. Kemudahan penggunaan aplikasi dapat mempengaruhi tingkat emosional pengguna. Yang dimaksud dari kemudahan ini adalah sesuai dengan konsep desain *UX (User Experience)* pada aplikasi ini. Kemudian pengukuran tingkat kemudahan penggunaan aplikasi virtual drum dapat diukur dengan kuisioner pada tahap pengujian nanti.

2. Kebutuhan Grafis

Daya tarik aplikasi berbanding lurus dengan kualitas grafis yang disajikan dalam aplikasi. Semakin bagus desain aplikasi maka semakin besar daya tarik pengguna untuk menggunakan aplikasi ini. Selain itu pengguna akan tertarik menggunakan aplikasi jika tampilan atau desain pada aplikasi menarik.

3. Kebutuhan Suara

Suara yang dikeluarkan pada aplikasi drum virtual harus sama dengan drum aslinya. Sehingga pengguna dapat merasakan seperti sedang bermain drum asli. Hal ini juga dapat mengacu pada konsep desain *UX (User Experience)* yaitu dapat memberikan pengalaman seperti nyata kepada pengguna.

3.4.4 Karakteristik Pengguna

Pengguna yang akan menggunakan aplikasi ini berjumlah satu orang, yaitu pengguna yang akan melakukan simulasi atau pembelajaran. Karakteristik pengguna tercantum dalam Tabel 3.1.

Tabel 3.5 Karakteristik pengguna

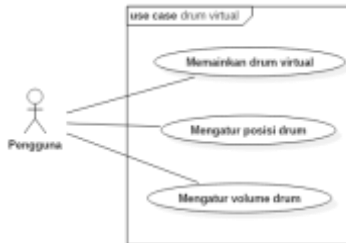
Nama Aktor	Tugas	Hak Akses Aplikasi	Kemampuan yang harus dimiliki
Pengguna	Pihak luar yang mencoba aplikasi	Menjalankan aplikasi	Tidak ada

3.5 Perancangan Sistem

Tahap perancangan dalam subbab ini dibagi menjadi beberapa bagian yaitu perancangan diagram kasus penggunaan, perancangan skenario kasus penggunaan, perancangan arsitektur, perancangan antarmuka pengguna, dan perancangan kontrol aplikasi.

3.5.1 Perancangan Diagram Kasus Penggunaan

Dalam aplikasi tugas akhir ini, terdapat tiga kasus penggunaan yang ada yaitu melakukan simulasi, menelusuri tempat-tempat haji, dan memperoleh informasi terkait objek-objek tertentu di area tempat tersebut. Rancangan kasus penggunaan dapat dilihat pada Gambar 3.12.



Gambar 3.12 Diagram kasus penggunaan

Penjelasan singkat dari masing-masing kasus penggunaan dapat dilihat pada Tabel 3.6.

Tabel 3.6 Skenario kasus penggunaan

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1	UC-001	Memainkan drum virtual dengan gerakan tangan	Pengguna dapat memainkan drum virtual menggunakan kamera Intel Realsense.
2	UC-002	Mengatur posisi drum	Pengguna dapat mengatur letak posisi komponen drum.
3	UC-003	Mengatur volume suara pada aplikasi	Pengguna dapat melakukan pengaturan suara aplikasi virtual drum.

3.5.2 Perancangan Skenario Kasus Penggunaan

Skenario kasus penggunaan dibagi menjadi tiga bagian sesuai dengan kasus penggunaan yang ada dan tercantum pada Tabel 3.7 sampai dengan

Tabel 3.9. Tabel tersebut berisi penjelasan skenario yang akan dilakukan ketika pengujian.

Tabel 3.7 Skenario Kasus Penggunaan Memainkan Drum Virtual dengan Tangan

Nama Kasus Penggunaan	Memainkan drum virtual dengan tangan.
Kode	UC-001
Deskripsi	Kasus penggunaan agar aktor dapat memukul drum virtual menggunakan gerakan tangan.
Aktor	Pengguna
Kondisi Awal	Pengguna sudah membuka aplikasi
Alur Kejadian Normal	<ol style="list-style-type: none">1. Pengguna memilih desain aplikasi drum virtual.2. Sistem menampilkan desain aplikasi sesuai desain yang dipilih pengguna.3. Pengguna dapat memainkan drum virtual dengan menggerakkan tangan maju ke arah komponen drum yang dipilih.

Tabel 3.8 Skenario Kasus Mengatur Posisi Alat Drum

Nama Kasus Penggunaan	Mengatur posisi drum
Kode	UC-002
Deskripsi	Kasus penggunaan agar aktor dapat mengatur letak koordinat x dan y alat musik drum.
Aktor	Pengguna
Kondisi Awal	Pengguna sudah membuka aplikasi
Alur Normal	<ol style="list-style-type: none">1. Pengguna memilih tombol posisi pada layar aplikasi.2. Sistem menonaktifkan suara dan alat musik drum siap untuk dirubah letaknya.3. Pengguna dapat mengubah posisi salah satu alat musik drum.

Tabel 3.9 Skenario Kasus Mengatur Volume Suara

Nama Kasus Penggunaan	Mengatur volume suara pada aplikasi
Kode	UC-003
Deskripsi	Kasus penggunaan agar pengguna dapat mengatur volume pada aplikasi.
Aktor	Pengguna
Kondisi Awal	Pengguna sudah membuka aplikasi.
Alur Normal	<ol style="list-style-type: none">1. Pengguna memilih tombol pengaturan volume pada aplikasi.2. Sistem menampilkan pengaturan volume aplikasi.3. Pengguna dapat mengubah rasio volume aplikasi dari 0 hingga 100.

(Halaman ini sengaja dikosongkan)

BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Di dalamnya mencakup proses penerapan dan pengimplementasian proses dan antarmuka yang mengacu pada rancangan yang telah dibahas sebelumnya.

4.1 Lingkungan Implementasi

Lingkungan implementasi tugas akhir dijelaskan di Tabel 4.1.

Tabel 4.1 Lingkungan implementasi perangkat lunak

Lingkungan Perangkat Keras	<p>Prosesor :</p> <ul style="list-style-type: none">- Intel(R) Core(TM) i5-5430M CPU @ 2.70GHz <p>Memori :</p> <ul style="list-style-type: none">- 8 GB- VGA Intel Iris Graphics 6100
Lingkungan Perangkat Lunak	<p>Sistem Operasi :</p> <ul style="list-style-type: none">- Microsoft Windows 10 Ultimate 64-bit <p>Perangkat Pengembang :</p> <ul style="list-style-type: none">- Corel Draw X5- Adobe Photoshop CS5- Unity 4.6.5- MonoDevelop v4.01

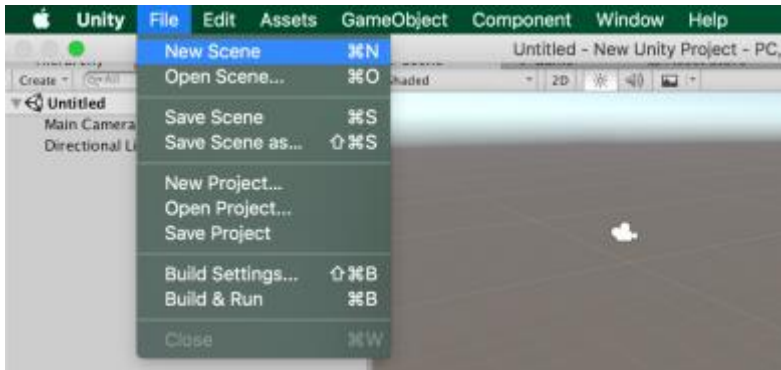
4.2 Implementasi Aplikasi Drum Virtual

Implementasi pembuatan desain aplikasi drum virtual dengan menampilkan *UI (User Interface)*. Pembuatan *UI* ini meliputi

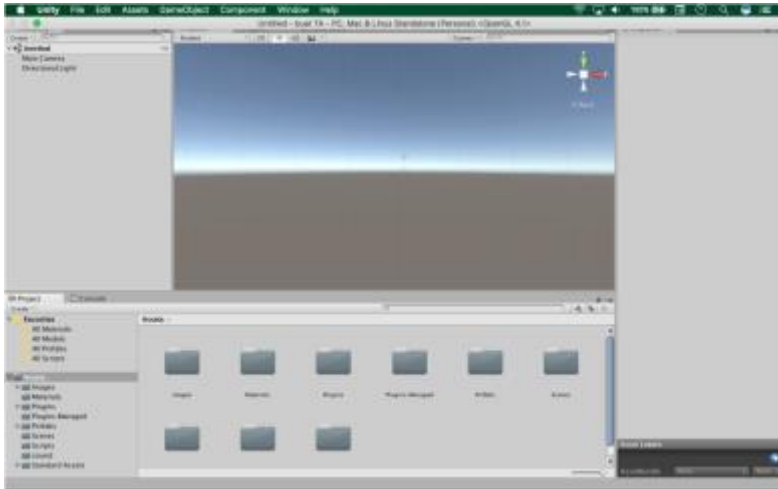
pembuatan komponen-komponen drum, tombol-tombol dan tulisan pada *scene Unity*.

4.2.1 Implementasi Pembuatan *Scene*

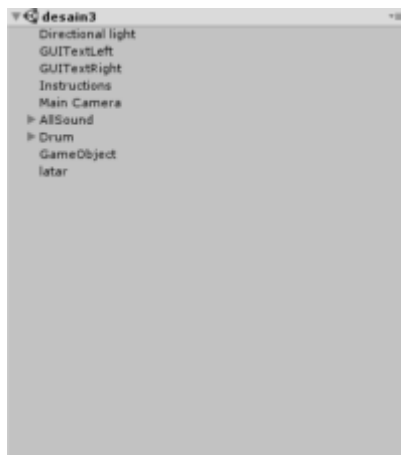
Untuk menambahkan *scene*, hal yang perlu dilakukan adalah menuju tab *file* lalu pilih *new scene*. Untuk lebih jelasnya, pembuatan *scene* baru ada di Gambar 4.1. Setelah *scene* terbuat, maka objek-objek yang telah kita miliki terdapat pada *project explorer* yang terletak di kiri bawah aplikasi *Unity* (Gambar 4.3). Untuk memasukkan objek-objek, berkas yang terdapat dalam proyek bisa dimasukkan dengan cara *drag and drop* berkas dari *map* yang terdapat dalam proyek ke *tab scene*. Untuk melihat properti objek mengenai posisi atau yang lainnya, pilih objek yang akan dilihat propertinya. Informasi properti terdapat pada label *Inspector*. Lebih jelasnya, ada pada Gambar 4.1.



Gambar 4.1 Penambahan *Scene*



Gambar 4.2 Tampilan *Scene* Baru



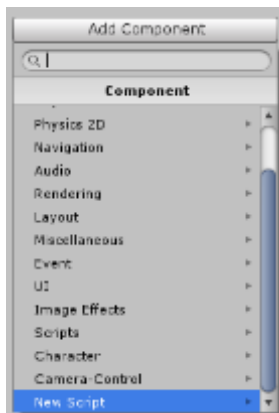
Gambar 4.3 Tampilan *Project Explorer*

4.2.2 Implementasi Pembuatan *Script*

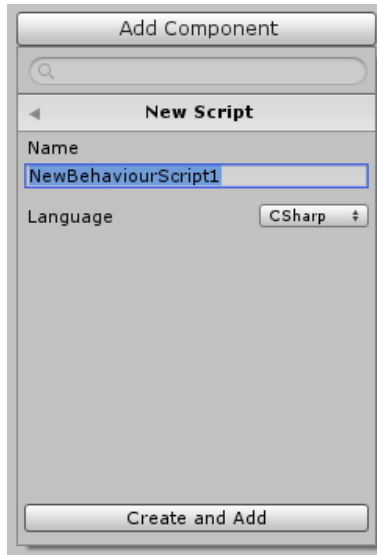
Untuk membuat *script*, dan mengaitkan terhadap objek, pilih objek yang akan dimasukkan *script*, lalu tambahkan komponen *script*. *Script* yang dibuat bisa dalam bentuk *javascript* atau C# Gambar 4.4, Gambar 4.5 dan Gambar 4.6).



Gambar 4.4 Pembuatan *script* (1)



Gambar 4.5 Pembuatan *script* (2)



Gambar 4.6 Pembuatan Script (3)

4.2.3 Implementasi Tampilan Drum

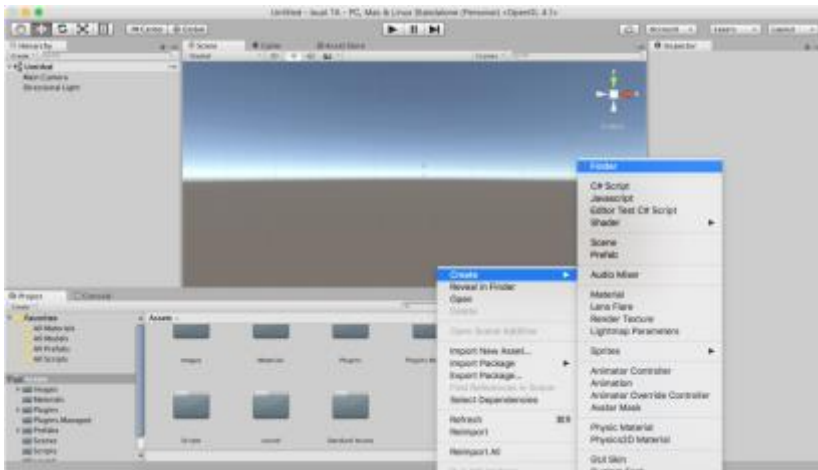
Untuk membuat tampilan drum pada aplikasi, perlu dilakukan import gambar dengan ekstensi (*.png*) atau (*.jpg*). Kelebihan gambar dengan format *.png* dari pada *.jpg* adalah gambar dengan ekstensi *.png* bisa transparan. Sementara itu, cara untuk mengimport gambar kemudian meletakkan di scene adalah sebagai berikut:

1. *Import* gambar Dengan format *.png* dan *.jpg* dengan cara pada Panel Project masuk ke dalam *folder Assets* seperti pada Gambar 4.7.
2. Buat folder baru pada folder *Assets* tersebut dengan cara klik kanan pada panel folder *Assets*, kemudian pilih menu *create*. Setelah itu pilih menu *folder* seperti pada Gambar 4.8.
3. Beri nama folder tersebut misalnya dengan nama *image*.
4. Kemudian masuk ke dalam folder *image*. Setelah itu klik kanan dan pilih menu *Import new Asset* seperti pada Gambar 4.9.

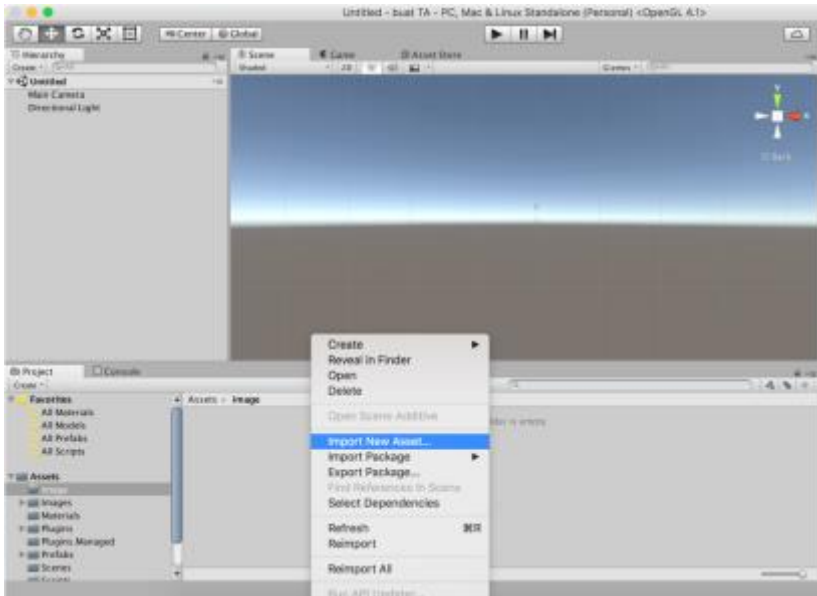
5. Kemudian pilih gambar dan setelah itu klik tombol import seperti pada Gambar 4.9.



Gambar 4.7 Folder Assets



Gambar 4.8 Cara Membuat Folder Baru



Gambar 4.9 Cara Import Asset Baru

4.2.4 Implementasi Pendeteksi Tumbukan Objek Drum dengan Tangan

Agar drum virtual dapat mendeteksi apakah komponen drum sudah dipukul dengan stik drum atau tidak, dapat dilakukan dengan fitur physics collision pada Unity. Pada Unity terdapat 3 *state collision* yang dapat dideteksi yaitu:

1. *OnCollisionStay* : merupakan status dimana objek satu saling bersentuhan dengan objek lainnya.
2. *OnCollisionEnter* : merupakan status dimana objek satu saling bertabrakan atau salah satu objek menembus objek lainnya.
3. *OnCollisionExit* : merupakan status dimana objek satu sudah tidak bersentuhan lagi dengan objek lainnya.

Kemudian untuk mengimplementasikan collision pada stik dan komponen drum dapat dilakukan dengan cara pembuatan script yang ditunjukkan *OnCollisionStay* pada Kode Sumber 4-1 beserta

penjelasan kode pada Tabel 4.1, *OnCollisionEnter* pada Kode Sumber 4-2 beserta penjelasan kode pada Tabel 4.3 dan *OnCollisionExit* pada Kode Sumber 4-3 beserta penjelasan kode pada Tabel 4.4 Penjelasan Kode Sumber 4-3.

```

1. void OnCollisionStay(Collision col)
2. {
3.     if (col.gameObject.name == "PalmCenter(Clone)"
        && check_collison == false ) {
4.
5.         if (hand_view.custom_mode == false && col.
            gameObject.transform.position.z < -
            40f && hand_view.stop == false) {
6.             drum.Play ();
7.             check_collison = true;
8.         }
9.     }

```

Kode Sumber 4-1 Saat Palm Berada di dalam Objek Komponen Drum

Tabel 4.2 Penjelasan Kode Sumber 4-1

No. Baris	Kegunaan
1	Fungsi OnCollisionStay untuk mendeteksi tumbukan jika ada objek yang berada tepat di dalam objek tersebut
3	Perintah yang hanya dilakukan jika objek yang berada di dalam adalah berupa objek dengan nama "PalmCntr(Clone)" dan status check_collison masih <i>false</i>
5	Perintah yang hanya dilakukan jika pengguna tidak mengaktifkan pemindahan objek dan posisi dalam koordinat z kurang dari -40f dan aplikasi tidak berhenti
6	Perintah yang digunakan untuk memutar suara
7	Perintah untuk mengaktifkan check_collison

```

1. void OnCollisionEnter(Collision col){
2.     drum_icon.color = Color.white;
3.     check_collison = false;
4. }

```

Kode Sumber 4-2 Saat Palm Menyentuh Komponen Drum

Tabel 4.3 Penjelasan Kode Sumber 4-2

No. Baris	Kegunaan
1	Fungsi OnCollisionEnter untuk mendeteksi tumbukan jika ada objek menyentuh objek tersebut
2	Perubahan warna komponen drum menjadi seperti semula
3	Menonaktifkan check_collison

```

1. void OnCollisionExit(Collision col){
2.     drum_icon.color = Color.white;
3.     check_collison = true;
4. }

```

Kode Sumber 4-3 Saat Palm Keluar dari Komponen Drum

Tabel 4.4 Penjelasan Kode Sumber 4-3

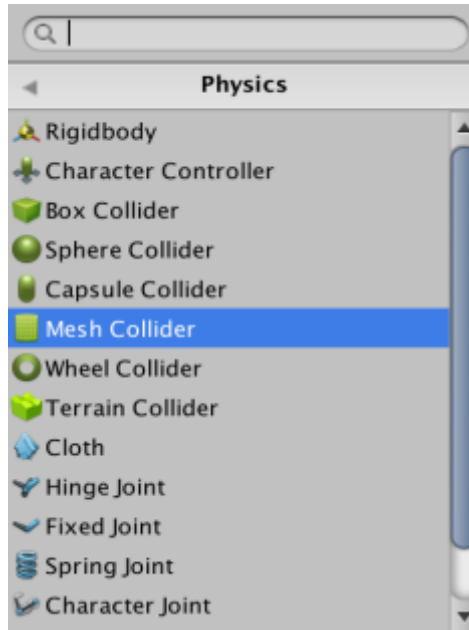
No. Baris	Kegunaan
1	Fungsi OnCollisionExit untuk mendeteksi tumbukan jika ada objek keluar dari objek tersebut
2	Perubahan warna komponen drum menjadi seperti semula
3	Mengaktifkan check_collison

Kode tersebut diletakkan pada setiap komponen drum. Tetapi sebelumnya perlu penambahan komponen *physics Mesh Collider* pada inspector *Unity*. Langkahnya adalah sebagai berikut.

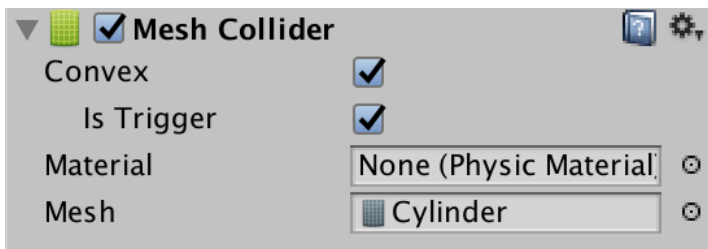
1. Pilih objek yang akan diberi *Mesh Collider*.
2. Pada panel *Inspector* Unity pilih tombol *Add Component* seperti pada Gambar 4.10.
3. Kemudian pilih *Component Physics* dan pilih *Mesh Collider* seperti pada Gambar 4.11.
4. Setelah itu centang *convex* dan *trigger* untuk mengaktifkan trigger.
5. Kemudian pada komponen stik (ujung stik) kita beri *sphere collider*. Langkahnya sama seperti langkah sebelumnya bedanya adalah pada stik ini Is Trigger tidak perlu dicentang pada Gambar 4.12.



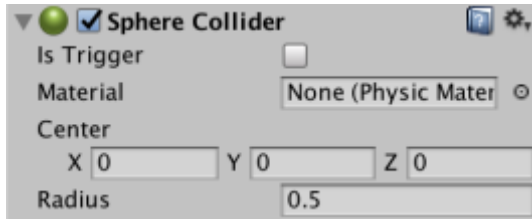
Gambar 4.10 Penambahan Komponen



Gambar 4.11 Pemilihan Mesh Collider



Gambar 4.12 Centang Is Trigger pada Mesh Collider

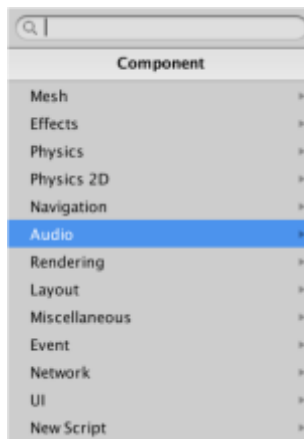


Gambar 4.13 Penambahan Sphere Collider pada Ujung Palm

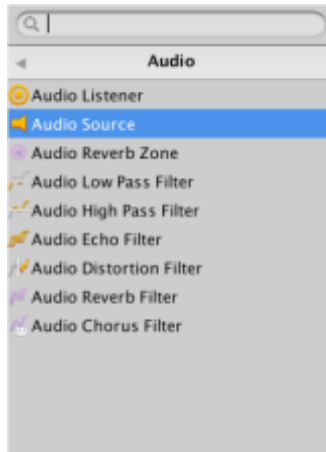
4.2.5 Implementasi Pemutaran Suara Drum

Saat stik drum menyentuh salah satu komponen drum maka yang seharusnya terjadi adalah akan terdapat suara drum. Pada *Unity* terdapat fitur *audio source* untuk memutar suara. Caranya adalah sebagai berikut.

1. Pilih objek yang akan diberi suara.
2. Kemudian pada panel Inspector pilih *Add Component*.
3. Setelah itu pilih menu *Audio* seperti pada Gambar 4.14.
4. Terakhir pilih *Audio Source* seperti pada Gambar 4.15.
5. Pilih *AudioClip* untuk memilih suara dengan format .wav

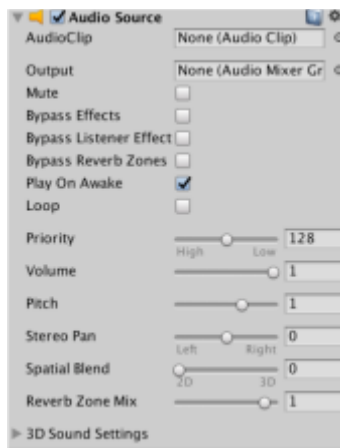


Gambar 4.14 Pemilihan Menu Audio



Gambar 4.15 Pemilihan Menu Audio Source

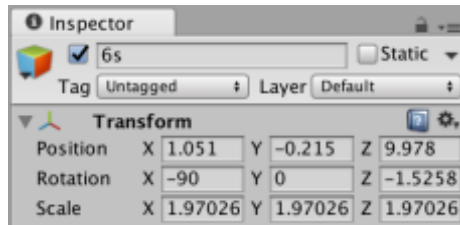
Agar suara tidak dimainkan saat pertama kali aplikasi dijalankan, togle *Play On Awake* jangan dicentang seperti pada Gambar 4.16.



Gambar 4.16 Toggle *Play On Awake* jangan dicentang

4.3 Implementasi Pemindahan Objek Drum yang Dapat Diatur oleh Pengguna

Setiap objek memiliki parameter koordinat titik x, y, dan z. Pada *Unity*, koordinat ini diatur di menu *Transform* pada *Inspector*. Sehingga pemindahan objek dapat diatur dengan fungsi *transform* yang tersedia. Untuk mengetahui letak koordinat objek pada *scene* dapat dilihat pada inspektor *Transform* seperti pada gambar Gambar 4.17.



Gambar 4.17 Inspector Transform

Terlihat bahwa objek dengan nama 6s memiliki posisi pada koordinat x = 1.051, y = -0.215, z = 9.978. Agar objek dapat berpindah posisinya maka perlu menambahkan script C# pada Kode Sumber 4-4 beserta penjelasan kode pada Tabel 4.5 dan kode proses pemindahan objek pada Kode Sumber 4-6 Variabel Global untuk Pendeteksi Jari beserta penjelasan kode pada Tabel 4.6.

```
1. //variabel global posisi tangan kiri
2. public float x_hand_left;
3. public float y_hand_left;
4. public float z_hand_left;
5.
6. //variabel global posisi tangan kanan
7. public float x_hand_right;
8. public float y_hand_right;
9. public float z_hand_right;
```

Kode Sumber 4-4 Variabel Global Posisi Palm Tangan Kanan dan Tangan Kiri

Tabel 4.5 Penjelasan Kode Sumber 4-4

No. Baris	Kegunaan
2	Koordinat x posisi Palm tangan kiri
3	Koordinat y posisi Palm tangan kiri
4	Koordinat z posisi Palm tangan kiri
7	Koordinat x posisi Palm tangan kanan
8	Koordinat y posisi Palm tangan kanan
9	Koordinat z posisi Palm tangan kanan

```

1. if (hand_view.custom_mode == true && (hand_view.ge
   sture_mode_left != "spreadfingers")) {
2.             if (hand_view.choose_component ==
   "null") {
3.                 hand_view.choose_component = t
   his.gameObject.name;
4.             }
5.             if (hand_view.choose_component ==
   this.gameObject.name) {
6.                 drum_icon.color = new Color(0.
   0f, 0.0f, 1.0f, 0.8f);
7.                 this.gameObject.transform.posi
   tion = new Vector3 (hand_view.x_hand_left, hand_vi
   ew.y_hand_left, hand_view.z_hand_left);
8.             }
9.         }

```

Kode Sumber 4-5 Proses Pemindahan Komponen Drum

Tabel 4.6 Penjelasan Kode Sumber 4-5

No. Baris	Kegunaan
1	Jika fitur pemindahan objek telah aktif dan gesture tangan kiri tidak sama dengan <i>spreadfinger</i> maka proses di dalamnya akan dilakukan
2	Jika komponen yang dipilih adalah masih belum ada, maka variabel <i>choose_component</i> adalah objek tersebut
5-6	Apabila objek yang dipilih telah aktif, maka komponen drum tersebut akan bewarna biru
7	Koordinat komponen drum akan mengikuti letak posisi tangan pengguna

Penjelasan dari Kode Sumber 4-5 adalah apabila *Palm* menyentuh drum maka objek drum tersebut akan di transformasikan sesuai koordinat jari.

4.4 Implementasi Pendeteksi Jari

Intel Realsense SDK memiliki fitur *finger tracking* yaitu fitur untuk mendeteksi gerakan tangan dan implementasi dari tangan ke dalam bentuk 3 Dimensi. Agar *PC/Laptop* dapat mendeteksi kamera *Intel Realsense* langkah pertama yang harus dilakukan adalah menginstal *DCM (Deep Camera Manager) Intel Realsense Driver* pada <https://software.intel.com/en-us/intel-realsense-sdk/download>. Kemudian langkah selanjutnya adalah menginstall *Intel Realsense SDK* pada <https://software.intel.com/en-us/intel-realsense-sdk/download>.

Berikut adalah kode untuk pendeteksi jari mulai dari inialisasi jari sampai transformasi jari dari Kode Sumber 4-6 sampai Kode Sumber 4-11 beserta penjelasan kode pada Tabel 4.7, Kode Sumber 4-7 Inialisasi Tangan dengan Intel Realsense **SDK** beserta penjelasan kode pada tabel Tabel 4.8, Kode Sumber 4-8 Inialisasi GameObject

Tulang dan Titik-titik **Joint** beserta penjelasan kode pada Tabel 4.9 Penjelasan Kode Sumber 4-8, Kode Sumber 4-9 Proses Pengubahan Data Koordinat Jari Pengguna yang Akan Ditampilkan di **Layar** beserta penjelasan kode pada Tabel 4.10, Kode Sumber 4-10 beserta penjelasan kode pada

Tabel 4.11.

```

1.  public GameObject JointPrefab; //Prefab untuk Joints
2.      public GameObject TipPrefab; //Prefab untuk Fingers Tips
3.      public GameObject BonePrefab; //Prefab untuk Bones
4.      public GameObject PalmCenterPrefab; //Prefab untuk Palm Center
5.
6.      public GUIText myTextLeft; //GUIText untuk tangan kiri
7.      public GUIText myTextRight; //GUIText untuk tangan kanan
8.
9.      private GameObject[][] myJoints; //Array dari Joint GameObjects
10.     private GameObject[][] myBones; //Array dari Bone GameObjects
11.
12.     private PXCMSHandData.JointData[][] jointData;
    //non-smooth joint values
13.     private PXCMSmoothing module smoother = null; //Smoothing module instance
14.     private PXCMSmoothing.Smoother3D[][] smoother3D = null; //smooth joint values
15.     private int weightsNum = 1; //smoothing factor
16.
17.     private PXCMSenseManager sm = null; //SenseManager Instance
18.     private pxcmStatus sts; //StatusType Instance
19.     private PXCMSHandModule handAnalyzer; //Hand Module Instance

```

```

20.     private int MaxHands = 2; //Max Hands
21.     private int MaxJoints = FXCMHandData.NUMBER_OF
    _JOINTS; //Maximal Joints
22.
23.     private Hashtable handList; //keep track of bod
    yside and hands for GUI text

```

Kode Sumber 4-6 Variabel Global untuk Pendeteksi Jari

Tabel 4.7 Penjelasan Kode Sumber 4-6

No. Baris	Kegunaan
1	Variabel untuk menampung gameObject <i>joint</i> jari
2	Variabel untuk menampung gameObject ujung jari
3	Variabel untuk menampung gameObject tulang jari
4	Variabel untuk menampung gameObject Palm
6	Text yang akan menampilkan gesture pada tangan kiri
7	Text yang akan menampilkan gesture pada tangan kanan
9	Array yang akan menampung Joint
10	Array yang akan menampung <i>bones</i> / tulang
12	Array yang akan menampung data-data pada setiap joint
17	Inisialisasi modul Intel Realsense
18	Variabel yang akan menyimpan status modul Intel Realsense
19	Variabel <i>Hand Module</i> pada Intel Realsense SDK
20	Variabel yang menampung jumlah maksimal tangan yang akan ditampilkan antara 1 atau 2
21	Variabel yang menampung berapa maksimal jumlah <i>joint</i> (titik sendi pada jari)

```

1.  void Start()
2.  {
3.      handList = new Hashtable();
4.
5.      /* Initialize a PXCMSenseManager instance */
6.      sm = PXCMSenseManager.CreateInstance();
7.      if (sm == null)
8.          Debug.LogError("SenseManager Initializat
ion Failed");
9.
10.     /* Enable hand tracking and retrieve an hand
module instance to configure */
11.     sts = sm.EnableHand();
12.     handAnalyzer = sm.QueryHand();
13.     if (sts != pxcmStatus.PXCM_STATUS_NO_ERROR)
14.         Debug.LogError("PXCSenseManager.EnableHa
nd: " + sts);
15.
16.     /* Initialize the execution pipeline */
17.     sts = sm.Init();
18.     if (sts != pxcmStatus.PXCM_STATUS_NO_ERROR)
19.         Debug.LogError("PXCSenseManager.Init: "
+ sts);
20.
21.     /* Retrieve the the DataSmoothing instance *
/
22.     sm.QuerySession().CreateImpl<PXCMsmoother>(o
ut smoother);
23.
24.     /* Create a 3D Weighted algorithm */
25.     smoother3D = new PXCMsmoother.Smoother3D[Max
Hands][];
26.
27.     /* Configure a hand - Enable Gestures and Al
erts */
28.     PXCMHandConfiguration hcfg = handAnalyzer.Cr
eateActiveConfiguration();
29.     if (hcfg != null)
30.     {
31.         hcfg.EnableAllGestures();

```

```

32.         hcfg.EnableAlert(PXCMHandData.AlertType.
    ALERT_HAND_NOT_DETECTED);
33.         hcfg.ApplyChanges();
34.         hcfg.Dispose();
35.     }
36.
37.     InitializeGameobjects();
38.
39. gesture_mode_right = "none";
40. gesture_mode_left = "none";
41.
42. }

```

Kode Sumber 4-7 Inisialisasi Tangan dengan Intel Realsense SDK

Tabel 4.8 Penjelasan Kode Sumber 4-7

No. Baris	Kegunaan
1	Fungsi yang dijalankan ketika pertama kali aplikasi dijalankan
5-8	Inisialisasi Intel Realsense SDK pada aplikasi
10-11	Pengaktifan <i>hand tracking</i>
1-18	Inisialisasi <i>pipeline</i>
21-22	Penerimaan <i>dataSmoothing</i> (perpindahan secara halus pada setiap objek)
24-25	Pembuatan algoritma 3D Weighted
27-37	Inisialisasi tangan, pengaktifan gesture dan pengaktifan peringatan

```

1.     void InitializeGameobjects()
2.     {
3.         myJoints = new GameObject[MaxHands][];
4.         myBones = new GameObject[MaxHands][];

```

```

5.         jointData = new PXCMSHandData.JointData[Max
    Hands][];
6.         for (int i = 0; i < MaxHands; i++)
7.         {
8.             myJoints[i] = new GameObject[MaxJoints
    ];
9.             myBones[i] = new GameObject[MaxJoints]
    ;
10.            smoother3D[i] = new PXCMSmoother.Smoot
    her3D[MaxJoints];
11.            jointData[i] = new PXCMSHandData.JointD
    ata[MaxJoints];
12.        }
13.        for (int i = 0; i < MaxHands; i++)
14.        for (int j = 0; j < MaxJoints; j++)
15.        {
16.            smoother3D[i][j] = smoother.Create
    3DWeighted(weightsNum);
17.            jointData[i][j] = new PXCMSHandData
    .JointData();
18.
19.            if (j == 1)
20.                myJoints[i][j] = (GameObject)I
    nstantiate(PalmCenterPrefab, Vector3.zero, Quatern
    ion.identity);
21.            else if (j == 21 || j == 17 || j =
    = 13 || j == 9 || j == 5)
22.                myJoints[i][j] = (GameObject)I
    nstantiate(TipPrefab, Vector3.zero, Quaternion.ide
    ntity);
23.            else
24.                myJoints[i][j] = (GameObject)I
    nstantiate(JointPrefab, Vector3.zero, Quaternion.i
    dentity);
25.
26.            if (j != 1)
27.                myBones[i][j] = (GameObject)In
    stantiate(BonePrefab, Vector3.zero, Quaternion.ide
    ntity);
28.        }
29.    }

```

Kode Sumber 4-8 Inisialisasi GameObject Tulang dan Titik-titik Joint

Tabel 4.9 Penjelasan Kode Sumber 4-8

No. Baris	Kegunaan
4	Inisialisasi <i>joint</i> dari jumlah maksimal tangan yang akan ditampilkan
5	Inisialisasi <i>bone</i> dari jumlah maksimal tangan yang akan ditampilkan
6	Inisialisasi data-data <i>joint</i> dari Intel Realsense SDK
7-11	Pemindahan data-data ke <i>joint</i> , <i>bone</i> dan <i>jointData</i>
15-30	Setelah selesai terinisialisasi maka langkah selanjutnya adalah data tersebut akan dijadikan <i>gameObject</i> berupa prefab <i>joint</i> , <i>bone</i> dan <i>palm</i>

```

1. //Smoother and Display the Joint Data
2.     void DisplayJoints()
3.     {
4.         for (int i = 0; i < MaxHands; i++)
5.         {
6.             for (int j = 0; j < 2; j++)
7.             {
8.                 if (jointData[i][j] != null && jointData[i][j].confidence == 100)
9.                 {
10.                    PXCMPPoint3DF32 smoothedPoint =
                        smoother3D[i][j].SmoothValue(jointData[i][j].positionWorld);
11.                    myJoints[i][j].SetActive(true)
12.                    ;
13.                    myJoints[i][j].transform.position = new Vector3(-
                        1 * smoothedPoint.x, jointData[i][j].positionWorld.y, -1*smoothedPoint.z) * 100f;
14.                    if (i==0 && j == 1) {
15.                        x_hand_left = -
                        1 * smoothedPoint.x*100;

```



```

15.             y_hand_left = smoothedPoin
    t.y*100;
16.             z_hand_left = -
    1 * smoothedPoint.z*100;
17.         }
18.         if (i==1 && j == 1) {
19.             x_hand_right = -
    1 * smoothedPoint.x*100;
20.             y_hand_right = smoothedPoi
    nt.y*100;
21.             z_hand_right = -
    1 * smoothedPoint.z*100;
22.         }
23.         jointData[i][j] = null;
24.     }
25.     else
26.     {
27.         myJoints[i][j].SetActive(false
    );
28.     }
29. }
30. }
31.
32.     for (int i = 0; i < MaxHands; i++)
33.         for (int j = 0; j < 2; j++)
34.         {
35.
36.             if (j != 21 && j != 0 && j != 1 &&
    j != 5 && j != 9 && j != 13 && j != 17)
37.                 UpdateBoneTransform(myBones[i]
    [j], myJoints[i][j], myJoints[i][j + 1]);
38.
39.                 UpdateBoneTransform(myBones[i][21]
    , myJoints[i][0], myJoints[i][2]);
40.                 UpdateBoneTransform(myBones[i][17]
    , myJoints[i][0], myJoints[i][18]);
41.
42.                 UpdateBoneTransform(myBones[i][5],
    myJoints[i][14], myJoints[i][18]);
43.                 UpdateBoneTransform(myBones[i][9],
    myJoints[i][10], myJoints[i][14]);
44.                 UpdateBoneTransform(myBones[i][13]
    , myJoints[i][6], myJoints[i][10]);

```

```
45.         UpdateBoneTransform(myBones[i][0],
    myJoints[i][2], myJoints[i][6]);
46.     }
47. }
```

Kode Sumber 4-9 Proses Pengubahan Data Koordinat Jari Pengguna yang Akan Ditampilkan di Layar

Tabel 4.10 Penjelasan Kode Sumber 4-9

No. Baris	Kegunaan
4	Perulangan yang dilakukan sampai maksimal tangan yang diinputkan
6	Perulangan yang dilakukan sampai maksimal jumlah joint (dalam kasus ini jumlah joint hanya satu yaitu <i>palm</i>)
10	Baris kode yang digunakan untuk menghaluskan perpindahan objek dari satu titik ke titik lainnya (dalam kasus ini adalah perpindahan <i>palm</i>)
11	Pengaktifan <i>gameObject palm</i>
12	Proses transformasi atau pemindahan objek <i>palm</i>
13-17	Pengambilan koordinat tangan kiri
18-22	Pengambilan koordinat tangan kanan
32-46	Transformasi data ke dalam <i>bone</i> agar tulang pada sendi dapat ditampilkan pada layar

```
1. //Update Bones
2. void UpdateBoneTransform(GameObject _bone, Gam
    eObject _prevJoint, GameObject _nextJoint)
3. {
4.
5.     if (_prevJoint.activeSelf == false || _nex
        tJoint.activeSelf == false)
```

```

6.         _bone.SetActive(false);
7.     else
8.     {
9.         _bone.SetActive(true);
10.
11.        // Update Position
12.        _bone.transform.position = ((_nextJoint
    t.transform.position - _prevJoint.transform.positi
    on) / 2f) + _prevJoint.transform.position;
13.
14.        // Update Scale
15.        _bone.transform.localScale = new Vecto
    r3(0.8f, (_nextJoint.transform.position - _prevJoi
    nt.transform.position).magnitude - (_prevJoint.tra
    nsform.position - _nextJoint.transform.position).m
    agnitude / 2f, 0.8f);
16.
17.        // Update Rotation
18.        _bone.transform.rotation = Quaternion.
    FromToRotation(Vector3.up, _nextJoint.transform.po
    sition - _prevJoint.transform.position);
19.    }
20. }

```

Kode Sumber 4-10 Trasformasi Bone agar dapat Ditampilkan di Layar

Tabel 4.11 Penjelasan Kode Sumber 4-10

No. Baris	Kegunaan
5-9	Perintah jika titik <i>joint</i> sebelum dan sesudah tidak aktif maka <i>bone</i> tidak ditampilkan, begitu sebaliknya
11-12	<i>Update</i> posisi pada <i>bone</i> untuk ditransformasikan
14-15	<i>Update</i> skala pada <i>bone</i> yang disesuaikan dengan letak titik pada <i>joint</i>
17-19	<i>Update</i> rotasi <i>bone</i> yang disesuaikan dengan kedua titik <i>joint</i>

```

1. //Close any ongoing Session
2.     void OnDisable()
3.     {
4.         if (smoother3D != null)
5.         {
6.             for (int i = 0; i < MaxHands; i++)
7.             {
8.                 if (smoother3D[i] != null)
9.                 {
10.                    for (int j = 0; j < 2; j++)
11.                    {
12.                        smoother3D[i][j].Dispose()
13.                    ;
14.                        smoother3D[i][j] = null;
15.                    }
16.                }
17.                smoother3D = null;
18.            }
19.
20.            if (smoother != null)
21.            {
22.                smoother.Dispose();
23.                smoother = null;
24.            }
25.
26.            if (sm != null)
27.            {
28.                sm.Close();
29.                sm.Dispose();
30.                sm = null;
31.            }
32.        }

```

Kode Sumber 4-11 Penonaktifan Semua *Joint* Intel Realsense SDK

4.5 Implementasi Penerjemah Pola Tangan atau Gesture

Selain memiliki *hand tracking*, Intel Realsense juga memiliki fitur untuk mendeteksi pola tangan atau gesture. Pola tangan yang dapat dideteksi oleh kamera Intel Realsense seperti *spreadfingers*,

first, tap, thumb_up, thumb_down dan lainnya. Pola tangan ini sangat berguna sebagai inputan dari pengguna. Contohnya adalah sebagai pengganti *mouse click* dapat diganti dengan gesture tap. *Gesture* ini mendeteksi pergerakan tangan maju kemudian mundur dengan cepat. Berikut adalah kode untuk menampilkan gesture pada Gambar 4.12 dan pada Kode Sumber 4-13.

```
1. //Display Gestures
2.     void DisplayGestures (PXCMLHandData.GestureData
   gestureData)
3.     {
4.         if (handList.ContainsKey (gestureData.hand
   Id)) {
5.             switch ((PXCMLHandData.BodySideType)han
   dList [gestureData.handId]) {
6.                 case PXCMLHandData.BodySideType.BODY_SI
   DE_LEFT:
7.                     myTextLeft.text = gestureData.name
   .ToString ();
8.                     gesture_mode_left = gestureData.na
   me.ToString ();
9.                     break;
10.                case PXCMLHandData.BodySideType.BODY_SI
   DE_RIGHT:
11.                    myTextRight.text = gestureData.nam
   e.ToString ();
12.                    gesture_mode_right = gestureData.n
   ame.ToString ();
13.                    break;
14.                }
15.            }
16.        }
```

Kode Sumber 4-12 Menampilkan *Gesture* atau Pola Tangan

```
1. this.gameObject.transform.position = new Vector3 (
   hand_view.x_hand_left, hand_view.y_hand_left, hand
   view.z_hand_left);
```

```

2.         if (hand_view.gesture_mode_left == "spread
fingers" || hand_view.gesture_mode_left == "tap"
3.             || hand_view.gesture_mode_left == "swi
pe_left" || hand_view.gesture_mode_left == "swipe_
right"
4.             || hand_view.gesture_mode_left == "wav
e" || hand_view.gesture_mode_left == "two_fingers_
pich_open")
5.             spreadfinger.SetActive (true);
6.         else
7.             spreadfinger.SetActive(false);
8.
9.         if (hand_view.gesture_mode_left == "fist")
10.            first.SetActive (true);
11.        else
12.            first.SetActive(false);
13.
14.        if (hand_view.gesture_mode_left == "thumb_
up")
15.            thumbUp.SetActive (true);
16.        else
17.            thumbUp.SetActive(false);
18.
19.        if (hand_view.gesture_mode_left == "thumb_
down")
20.            thumbDown.SetActive (true);
21.        else
22.            thumbDown.SetActive(false);
23.
24.        if (hand_view.gesture_mode_left == "v_sign
")
25.            vSign.SetActive (true);
26.        else
27.            vSign.SetActive(false);
28.
29.        if (hand_view.gesture_mode_left == "full_p
inch")
30.            fullPinch.SetActive (true);
31.        else
32.            fullPinch.SetActive(false);
33.

```

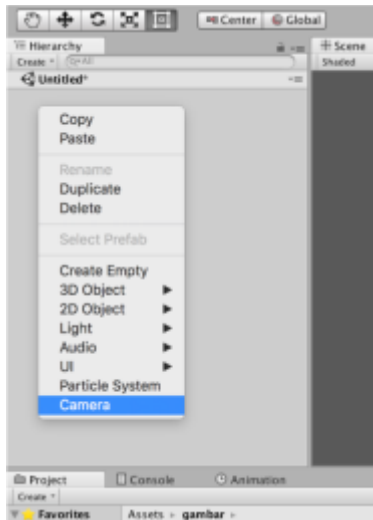
```
34.         if (hand_view.gesture_mode_left == "none")
35.             none.SetActive (true);
36.         else
37.             none.SetActive(false);
```

Kode Sumber 4-13 Menampilkan Gambar Pola Tangan pada Layar

4.6 Implementasi Pembuatan Kamera pada Unity

Agar komponen dapat terlihat pada layar desktop maka perlu ditambahkan kamera pada Unity. Langkah pembuatan kamera pada Unity adalah sebagai berikut:

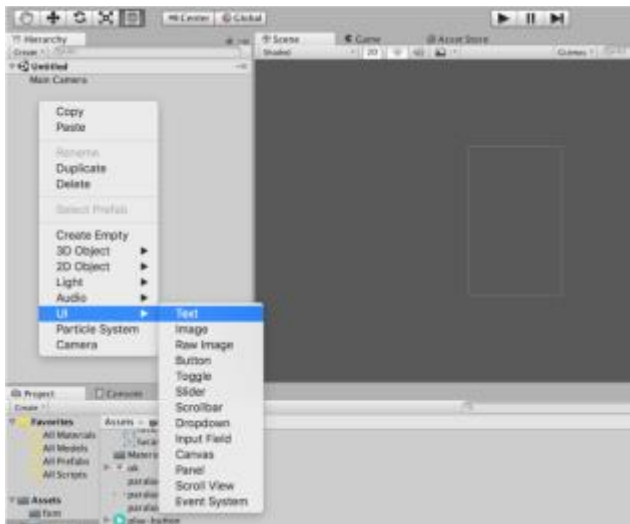
1. Pada panel *hierarchy* klik kanan *mouse* kemudian pilih *Create Empty* seperti pada Gambar 4.18.
2. Kemudian pilih Objek baru tersebut, pada panel Inspector tambahkan komponen dengan cara klik *Add Component*
3. Pilih menu *rendering* kemudian pilih *camera*.



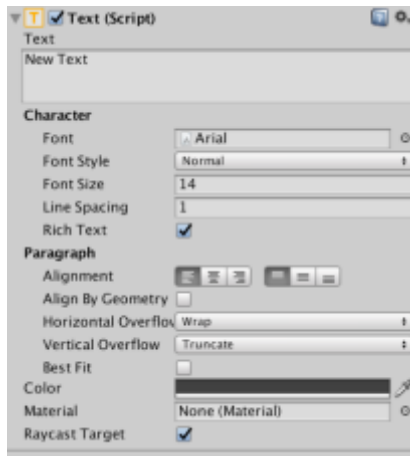
Gambar 4.18 Cara Pembuatan Kamera

4.7 Implementasi Menampilkan Text pada Scene

Aplikasi ini nantinya akan menampilkan keterangan kepada pengguna berupa peringatan apabila kamera Intel Realsense tidak menjangkau tangan pengguna. Terdapat dua kemungkinan pada peringatan ini yaitu yang pertama adalah tangan pengguna terlalu jauh dari jangkauan kamera, kemudian yang kedua adalah tangan pengguna terlalu dekat dengan jangkauan kamera. Hal ini disebabkan karena kamera Intel Realsense memiliki jangkauan hingga 25 cm dari depan kamera agar dapat mendeteksi tangan pengguna. Untuk pembuatan text pada Unity dapat di lihat pada Gambar 4.19 serta *inspector text* pada Gambar 4.20.



Gambar 4.19 Pembuatan *text* pada Unity



Gambar 4.20 Inspector *text* pada Unity

Text pada Unity memiliki beberapa fitur untuk mengatur *text* tersebut. Pada tab *character*, fitur font digunakan untuk mengganti tipe font apa yang akan dipakai seperti pada Gambar 4.20. Fitur font style digunakan untuk mengganti gaya font seperti *normal*, *italic* dan *bold*. Fitur *line spacing* digunakan untuk mengatur jarak antara kalimat satu dengan kalimat lainnya.

4.8 Implementasi Penghitungan Task Accuracy

Perhitungan *task accuracy* adalah perhitungan yang dilakukan untuk mengukur seberapa akurasi pengguna mengerjakan *task*. berikut adalah *script* dari perhitungan *task accuracy* pada Kode Sumber 4-14.

```

1. if (collisionInfo.gameObject.name == nameOfObject)
   {
2.           //Debug.Log (this.gameObject.transform
   .position.y);
3.           float hasil = 0;
4.           if (gameObject.transform.position.y >= nilai_tengah)
5.               hasil = (nilai_atas - (gameObject.transform.position.y)) / (nilai_atas - nilai_tengah)
               *100f;

```

```

6.         else if(gameObject.transform.position.
           y<=nilai_tengah)
7.             hasil = (gameObject.transform.posi
           tion.y - (nilai_bawah) )/(nilai_tengah - nilai_baw
           ah)*100f;
8.         //Debug.Log (hasil);
9.         if (hasil>=0 && hasil<=100) {
10.            if (nameOfObject == "hihat" && col
           lisionInfo.gameObject.name == "hihat") {
11.                speedOfHint.check_hihat = true
           ;
12.                speedOfHint.percent_of_hihat =
           hasil;
13.                if (speedOfHint.check_hihat_an
           d_hit == true && speedOfHint.check_hihat_and_hit_t
           emp == false) {
14.                    speedOfHint.count = speedO
           fHint.count + 1;
15.                    speedOfHint.check_hihat_an
           d_hit_temp = true;
16.                }
17.            }
18.            if (nameOfObject == "snare" && col
           lisionInfo.gameObject.name == "snare") {
19.                speedOfHint.check_snare = true
           ;
20.                speedOfHint.percent_of_snare =
           hasil;
21.                if (speedOfHint.check_snare_an
           d_hit == true && speedOfHint.check_snare_and_hit_t
           emp == false) {
22.                    speedOfHint.count = speedO
           fHint.count + 1;
23.                    speedOfHint.check_snare_an
           d_hit_temp = true;
24.                }
25.            }
26.            if (nameOfObject == "kick" && coll
           isionInfo.gameObject.name == "kick") {
27.                speedOfHint.check_kick = true;
28.                speedOfHint.percent_of_kick =
           hasil;

```

```

29.         if (speedOfHint.check_kick_and
    _hit == true && speedOfHint.check_kick_and_hit_tem
    p == false) {
30.             speedOfHint.count = speedO
    fHint.count + 1;
31.             speedOfHint.check_kick_and
    _hit_temp = true;
32.         }
33.
34.     }
35. }
36. }

```

Kode Sumber 4-14 Implementasi Perhitungan Task Accuracy

4.9 Implementasi Penghitungan Task Complete

Proses perhitungan *task complete* dilakukan untuk menghitung berapa lama pengguna menyelesaikan tugas berupa *not-not*. Implementasi yang paling sederhana adalah dengan cara menggunakan fungsi waktu pada Unity yaitu ketika pengguna memulai mengerjakan *task* berarti waktu dimulai, dan ketika pengguna menyelesaikan *task* waktu akhir dicatat untuk dihitung berapa lama pengguna menyelesaikan *task* tersebut. Berikut adalah implementasi perhitungan *task complete* pada Kode Sumber 4-15.

```

1. timeHit = Time.time.ToString();
2. if (hand_view.count >= hand_view.total_count && st
    op_record == false) {
3.     hand_view.stop = true;
4.     int total_hit = total_hit_true + total_hit_fal
    se;
5.     float percent_hit = percent_hit_true / total_h
    it;
6.     float percent = hand_view.percent / hand_view.
    total_true;
7.     if(total_hit!=0)
8.         savePreset (this.gameObject.name, "total=" + t
    otal_hit + ", false=" + total_hit_false

```

```

9.         +", true=" + total_hit_true +", percent="+
percent_hit+" , total_error="+hand_view.total_error
10.         +", total_hit_true="+hand_view.total_true+
", percent_total="+percent);
11.     stop_record = true;
12. }

```

Kode Sumber 4-15 Implementasi Perhitungan Task Complete

4.10 Implementasi Penghitungan Task Error

Proses perhitungan *task error* dilakukan untuk menghitung jumlah kesalahan pengguna saat mengerjakan *task*. Hal yang perlu dilakukan adalah menghitung jumlah drum yang seharusnya tidak dipukul oleh pengguna. Berikut adalah implementasi *script* pada Unity pada Kode Sumber 4-16.

```

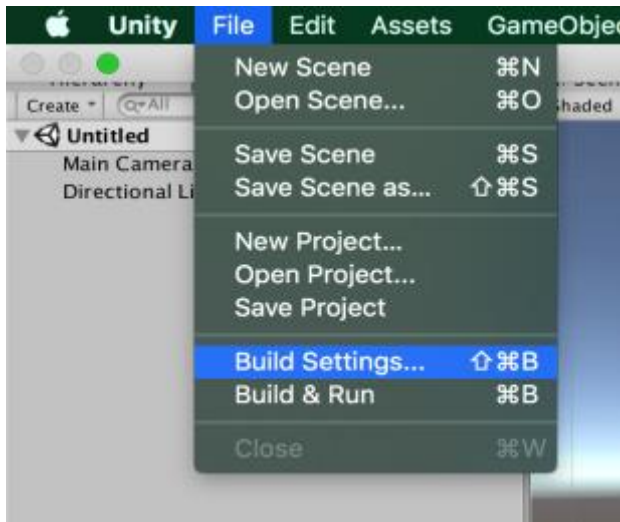
1. total_hit_false = total_hit_false + 1;
2. hand_view.total_error = hand_view.total_error+1;
3. drum_icon.color = new Color(1.0f, 0.0f, 0.0f, 0.8f
);

```

Kode Sumber 4-16 Implementasi Task Error

4.11 Pembuatan Proyek

Untuk membuat proyek, hal yang perlu dilakukan adalah memilih *tab "file"*, lalu pilih menu "*build settings...*", proses bisa dilihat pada Gambar 4.21.



Gambar 4.21 Cara Membuat Proyek

(halaman ini sengaja dikosongkan)

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas hal mengenai uji coba dan evaluasi aplikasi drum virtual dengan menggunakan Intel Realsense. Uji coba ini digunakan untuk menguji baik dari sisi perangkat lunak dan perangkat keras.

5.1 Lingkungan Uji Coba

Pada proses uji coba ini, lingkungan dibedakan menjadi lingkungan perangkat keras dan perangkat lunak. Berikut ini akan dijelaskan mengenai tiap-tiap lingkungan uji coba aplikasi.

5.1.1 Lingkungan Perangkat Keras

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam rangka uji coba perangkat lunak ini dicantumkan pada Tabel 5.1.

Tabel 5.1 Lingkungan Perangkat Keras

No.	Deskripsi
1	Intel(R) Core(TM) i5-5430M CPU @ 2.70GHz Memori : 8.00 GB

5.1.2 Lingkungan Perangkat Lunak

Deskripsi perangkat lunak untuk proses uji coba dapat dilihat pada Tabel 5.2.

Tabel 5.2 Lingkungan Perangkat Lunak

No.	Deskripsi
1	Sistem Operasi Windows 10 Pro 64-bit.
2	<i>Library</i> Intel Realsense SDK
3	Aplikasi Unity

5.2 Pengujian Fungsionalitas

Untuk mengetahui kesesuaian keluaran dari tiap tahap dan langkah penggunaan fitur terhadap skenario yang dipersiapkan, maka dibutuhkan pengujian fungsionalitas. Penjabaran skenario dan hasil uji coba fungsionalitas yang dilakukan terhadap perangkat lunak yang dibangun akan dijabarkan pada subbab ini.

5.2.1 Cara Menjalankan Aplikasi

Aplikasi ini membutuhkan perangkat Intel Realsense. Untuk memakai perangkat tersebut, dibutuhkan aplikasi kalibrasi bernama SDK Intel Realsense. Aplikasi tersebut tersedia beserta perangkat Intel Realsense atau pengguna bisa mengunduh pada alamat <https://software.intel.com/en-us/intel-realsense-sdk>. Setelah pemasangan berhasil, maka aplikasi yang telah dibuat tadi bisa langsung dijalankan.

5.2.2 Skenario Uji Coba Fungsionalitas

Subbab ini akan menjelaskan beberapa skenario uji coba perangkat lunak berdasarkan metode kotak hitam sebagai dasar tolok ukur keberhasilan. Pengujian fungsionalitas terhadap fitur-fitur dari aplikasi ini akan dijelaskan sebagai berikut:

- Uji memainkan drum virtual menggunakan kedua tangan.
- Uji coba mengatur posisi alat drum menggunakan satu tangan.

- Uji coba mengatur volume pada aplikasi menggunakan satu tangan.

Berdasarkan daftar pengujian yang telah disebutkan, dibuat beberapa skenario yang dilakukan pada setiap pengujian tersebut. Penjelasan mengenai cara dan hasil pengujian fungsionalitas perangkat lunak dibahas pada subbab Hasil Uji Coba.

5.2.3 Skenario Uji Coba Desain UX (*User Experience*)

Uji coba dilakukan dengan cara menggunakan *Experiment Task* dengan Latin Square 3 x 3. Terdapat 6 pengguna kemudian pengguna tersebut dibagi menjadi 3 kelompok (I, II, III), masing-masing kelompok 2 orang. Kemudian kelompok tersebut memainkan urutan desain aplikasi drum virtual sesuai baris pada tabel Latin Square di bawah ini.

Tabel 5.3 Experiment Task

	Uji Coba Ke-1	Uji Coba Ke-2	Uji Coba Ke-3
Kelompok I	A	B	C
Kelompok II	B	C	A
Kelompok III	C	B	A

Dari skenario uji coba desain UX (*User Experiment*) akan dibandingkan dari ketiga desain (Desain A, B, C) tersebut. Hasil uji coba didapatkan dari kuisioner pengguna berupa rating antara 1 – 6 berupa kemudahan, kenaturalan dan kenyamanan penggunaan aplikasi seperti pada Tabel 3.2. Selain dari kuisioner tersebut akan dihitung juga *task completion time*, *accuracy* dan *error rate* seperti pada Tabel 3.3 yang berupa *dependent variable* untuk mengukur secara objektif penggunaan aplikasi drum virtual.

5.2.4 Hasil Uji Coba

Pada subbab ini dijelaskan secara detail mengenai skenario yang dilakukan dan hasil yang didapatkan dari pengujian fungsionalitas perangkat lunak yang dibangun. Penjelasan disajikan dengan menampilkan kondisi awal, masukan, keluaran, hasil yang dicapai, dan kondisi akhir. Berikut ini merupakan penjabaran skenario dan hasil pengujian yang dicapai pada tiap-tiap fungsionalitas perangkat lunak.

5.2.4.1 Uji Coba Memainkan Aplikasi Drum Virtual

Uji coba memainkan aplikasi drum virtual menggunakan Intel Realsense adalah pengujian fungsi aplikasi dalam memainkan drum menggunakan kedua tangan. Dengan aplikasi yang sudah diintegrasikan dengan Intel Realsense pengguna bisa memainkan drum virtual dengan mengikuti *not-not* yang sedang berjalan seperti pada aplikasi Guitar Hero. *Not-not* tersebut merupakan *task* yang harus dimainkan oleh pengguna sampai berhasil. Task yang harus diselesaikan berjumlah 4 buah seperti di bawah ini.

1. Tugas Pertama

Tugas pertama yang akan dimainkan adalah Hi-hat saja. Hi-hat dinotasikan dengan "x" simbol di atas garis atas. Angka 1-4 yang adalah untuk menunjukkan bagaimana menghitung ini dengan suara keras.



Gambar 5.1 *Not-Not* Pada Task Pertama

2. Tugas Kedua

Tugas kedua akan memainkan *bass drum* yang dimainkan pada ketukan satu dan tiga. Fokus untuk menjaga empat ketukan *hi-hat* secara stabil, dan kemudian tambahkan dalam *bass drum*. *Bass drum* harus disinkronkan dengan sempurna dengan hi-hat ini.



Gambar 5.2 Not-Not pada Task Kedua

3. Tugas Ketiga

Tugas ketiga membawa *snare drum* ke dalam campuran. Di sini Anda tidak akan lagi bermain *bass drum*, tapi Anda akan memukul *snare drum* pada ketukan dua dan empat. Kedua dan keempat tuduhan menjadi satu tembakan gabungan.



Gambar 5.3 Not-Not pada Task Ketiga

4. Tugas Keempat

Dalam tugas keempat, Anda akan belajar untuk memainkan tiga pola pertama secara simultan. *Hi-hat* akan terus pada semua empat hitungan, *bass drum* dan *snare drum*. Fokus untuk tetap stabil. Banyak pemula akan memiliki kecenderungan untuk mulai bermain sedikit berombak.



Gambar 5.4 Not-Not pada Task Keempat

Selain itu, pengguna juga mendapatkan pengalaman berbeda seperti seolah-olah memainkan drum pada aslinya. Uji coba ini berfungsi untuk mengetahui desain manakah yang tepat untuk aplikasi drum virtual dari Desain A, B dan C. Karena terdapat 3 desain aplikasi maka untuk pengujian fungsionalitas aplikasi ini juga dibagi menjadi

3 tipe dengan 3 kelompok (I, II dan III). Untuk kelompok I akan menjalankan aplikasi dengan urutan desain aplikasi drum virtual A ke B dan ke C. Untuk kelompok II akan menjalankan aplikasi dengan urutan desain aplikasi drum virtual B ke C dan ke A. Untuk kelompok III akan menjalankan aplikasi dengan urutan desain aplikasi drum virtual C ke B dan ke A.

Tabel 5.4 Hasil Uji Coba Memainkan Drum Virtual

ID	UJ-UC-001
Nama	Uji Coba Memainkan aplikasi drum virtual
Tujuan uji coba	Menyelesaikan keempat tugas dengan memainkan drum sesuai not pada layar
Kondisi awal	Pengguna sudah membuka aplikasi
<i>Skenario 1</i>	<i>Pengguna memainkan drum virtual dengan gerakan tangan sesuai tugas pertama</i>
Masukan	Posisi kedua tangan pengguna dan gerakan tangan pengguna.
Keluaran yang diharapkan	Pengguna dapat menyelesaikan tugas pertama
Hasil uji coba	Berhasil.
Kondisi akhir	Pengguna berhasil menjalankan tugas pertama
<i>Skenario 2</i>	<i>Pengguna memainkan drum virtual dengan gerakan tangan sesuai tugas kedua</i>
Masukan	Posisi kedua tangan pengguna dan gerakan tangan pengguna.
Keluaran yang diharapkan	Pengguna dapat menyelesaikan tugas kedua
Hasil uji coba	Berhasil.
Kondisi akhir	Pengguna berhasil menjalankan tugas kedua

ID	UJ-UC-001
<i>Skenario 3</i>	<i>Pengguna memainkan drum virtual dengan gerakan tangan sesuai tugas ketiga</i>
Masukan	Posisi kedua tangan pengguna dan gerakan tangan pengguna.
Keluaran yang diharapkan	Pengguna dapat menyelesaikan tugas ketiga
Hasil uji coba	Berhasil.
Kondisi akhir	Pengguna berhasil menjalankan tugas ketiga
<i>Skenario 4</i>	<i>Pengguna memainkan drum virtual dengan gerakan tangan sesuai tugas keempat</i>
Masukan	Posisi kedua tangan pengguna dan gerakan tangan pengguna.
Keluaran yang diharapkan	Pengguna dapat menyelesaikan tugas keempat
Hasil uji coba	Berhasil
Kondisi akhir	Pengguna dapat menyelesaikan tugas keempat

5.2.4.2 Uji Coba Pemindahan Alat Musik Drum

Uji coba pemindahan alat musik drum digunakan untuk menyesuaikan tata letak sesuai keinginan pengguna. Pengguna dapat memindahkan tata letak komponen drum dengan cara memilih menu pemindahan alat musik drum kemudian pilih komponen drum yang akan dipindahkan dengan cara menggenggam tangan. Setelah itu pengguna mengarahkan komponen tersebut sesuai keinginan. Apabila sudah selesai memindahkan komponen alat musik, pengguna melepaskan genggam tangan tersebut. Fitur ini menggunakan kecanggihan pendeteksi pola tangan pada Intel Realsense.

Tabel 5.5 Hasil Uji Coba Pemindahan Alat Musik Drum

ID	UJ-UC-002
Nama	Uji Coba pemindahan alat musik drum
Tujuan uji coba	Memindahkan komponen drum sesuai keinginan pengguna.
Kondisi awal	Pengguna sudah membuka aplikasi
<i>Skenario 1</i>	<i>Pengguna memilih menu pemindahan objek pada layar aplikasi menggunakan gerakan tangan kanan</i>
Masukan	Pengguna menggerakkan posisi tangan ke menu pemindahan objek
Keluaran yang diharapkan	Menu pemindahan komponen alat musik drum telah aktif
Hasil uji coba	Berhasil
Kondisi akhir	Tampilan menu posisi pemindahan komponen drum telah berubah (aktif)
<i>Skenario 2</i>	<i>Pengguna menggenggam komponen drum yang akan dipindahkan</i>
Masukan	Pengguna menggerakkan posisi tangan kemudian menggenggam komponen drum yang akan dipindahkan.
Keluaran yang diharapkan	Komponen drum yang akan dipindahkan telah terpilih.
Hasil uji coba	Berhasil
Kondisi akhir	Tampilan komponen drum yang akan dipindahkan telah berubah
<i>Skenario 3</i>	<i>Pengguna memindahkan komponen drum dengan cara menggerakkan tangan</i>

ID	UJ-UC-002
Masukan	Pengguna menggerakkan posisi tangan untuk memindahkan komponen drum sesuai keinginan pengguna
Keluaran yang diharapkan	Komponen drum telah berpindah dari posisi semula
Hasil uji coba	Berhasil
Kondisi akhir	Komponen drum berpindah dari posisi awal ke posisi yang diinginkan oleh pengguna
Skenario 4	<i>Pengguna melepaskan genggamannya sebagai akhir dari proses pemindahan komponen drum</i>
Masukan	Pengguna melepaskan genggamannya
Keluaran yang diharapkan	Komponen drum yang awalnya terpilih menjadi tidak terpilih
Hasil uji coba	Berhasil
Kondisi akhir	Komponen drum tidak terpilih lagi

5.2.4.3 Uji Coba Pengaturan Volume Suara pada Aplikasi

Uji coba pengaturan volume suara berfungsi untuk mengatur keras atau kecilnya volume suara yang dapat dilakukan oleh pengguna menggunakan gerakan tangan dan fungsi gesture sebagai inputannya.

Tabel 5.6 Uji Coba Pengaturan Volume Suara

ID	UJ-UC-003
Nama	Uji Coba Pengaturan Volume Suara
Tujuan uji coba	Melakukan pengaturan volume suara sesuai keinginan pengguna
Kondisi awal	Pengguna membuka pengaturan volume suara

ID	UJ-UC-003
<i>Skenario 1</i>	<i>Pengguna menggenggam tombol slider pengaturan volume suara</i>
Masukan	Pola gesture tangan pengguna dengan pola <i>full_pinch</i>
Keluaran yang diharapkan	Tombol slider pada pengaturan volume suara telah aktif.
Hasil uji coba	Berhasil.
Kondisi akhir	Tombol slider berubah menjadi aktif
<i>Skenario 2</i>	<i>Pengguna menggerakkan slider pengaturan volume suara.</i>
Masukan	Gerakan tangan pengguna ke kanan (untuk membesarkan volume) atau ke kiri (untuk mengecilkan suara)
Keluaran yang diharapkan	Pengguna selesai mengatur volume suara sesuai keinginan.
Hasil uji coba	Berhasil.
Kondisi akhir	Suara berubah menjadi besar atau kecil sesuai keinginan pengguna.

5.3 Pengujian Non-Fungsionalitas

Pengujian non-fungsionalitas dilakukan untuk mengetahui bagaimana hasil keluaran selain dari sisi non-fungsionalitas sistem terhadap skenario yang dipersiapkan. Berikut ini penjabaran skenario dan hasil uji coba non-fungsionalitas yang dilakukan terhadap perangkat lunak yang dibangun.

5.3.1 Skenario Uji Coba Non-Fungsionalitas

Subbab ini akan menjelaskan beberapa skenario uji coba perangkat lunak dengan mengadakan uji coba kepada beberapa orang yang belum pernah melakukan proses haji sebagai dasar tolak ukur

keberhasilan. Pengujian non-fungsionalitas yang terdapat pada aplikasi dijabarkan sebagai berikut:

- Uji coba kemudahan penggunaan aplikasi
- Uji coba kenaturalan penggunaan aplikasi.
- Uji coba kenyamanan pengoperasian aplikasi.

Dari daftar pengujian yang telah disebutkan, akan dibuat beberapa skenario yang dilakukan pada setiap pengujian tersebut. Penjelasan mengenai cara dan hasil pengujian non-fungsionalitas perangkat lunak dibahas pada subbab 5.3.2.

5.3.2 Hasil Uji Coba

Subbab ini akan menjelaskan secara detail mengenai skenario yang dilakukan dan hasil yang didapat dari pengujian non-fungsionalitas perangkat lunak yang telah dibangun. Penjelasan akan ditampilkan berdasarkan hasil kuisioner yang telah diisi oleh partisipan setelah menggunakan aplikasi.

5.3.2.1 Daftar Penguji Perangkat Lunak

Pada subbab ini ditunjukkan daftar pengguna yang bertindak sebagai penguji coba aplikasi yang telah dibangun. Daftar nama penguji aplikasi ini dapat dilihat pada Tabel 5.10.

Tabel 5.7 Daftar nama penguji coba aplikasi

Responden	Kelompok	Pekerjaan
Rahmat Rijal	I	Mahasiswa
Novita Retno P. L	I	Mahasiswa
Ibnu Prayogi	II	Mahasiswa
Relaci Apilia I	II	Mahasiswa
Mardiana Sekarsari	III	Mahasiswa
Hanif Sudira	III	Mahasiswa

5.3.2.2 Hasil Uji Coba Kemudahan Penggunaan Aplikasi

Uji coba kemudahan aplikasi adalah uji coba yang dilakukan untuk mengukur seberapa mudah aplikasi setiap desain digunakan. Hasil uji coba akan dipaparkan pada Tabel 5.8, sedangkan detail pengujian pada tiap aspeknya akan ditampilkan pada Tabel 5.9.

Tabel 5.8 Rata-Rata Hasil Uji Kemudahan Penggunaan Aplikasi

Desain	Rata-rata Nilai
Desain Tipe A	4.33
Desain Tipe B	3.66
Desain Tipe C	4.50

Tabel 5.9 Detail Penilaian Kemudahan Penggunaan Aplikasi Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	4	4	5
Novita Retno P. L	3	4	6
Ibnu Prayogi	4	3	4
Relaci Apilia I	5	4	4
Mardiana Sekarsari	5	3	4
Hanif Sudira	5	4	4
TOTAL	26	22	27

Setelah dilakukan pengujian, terlihat pada Tabel 5.8 desain yang memiliki nilai kemudahan tertinggi adalah desain tipe C dengan nilai 4.5, hal ini dikarenakan menurut pengguna desain tipe C tata letaknya hampir sama dengan drum aslinya sehingga untuk memainkannya lebih mudah. Kemudian desain yang mudah setelah desain C adalah desain A dengan nilai 4.33, hal ini dikarenakan menurut beberapa pengguna adalah tata letak yang sejajar dapat memudahkan untuk memainkan drum tersebut. Dan yang terakhir adalah desain B dengan nilai 3.66, menurut pengguna desain B kurang

nyaman karena tata letaknya yang melingkar sehingga dapat membingungkan pengguna saat memainkannya.

5.3.2.3 Hasil Uji Coba Kenaturalan Penggunaan Aplikasi

Uji coba kenaturalan aplikasi adalah pengujian yang dilakukan untuk menguji seberapa natural aplikasi setiap desain A, B atau C. Hasil uji coba akan dipaparkan secara lengkap pada Tabel 5.10, Sedangkan untuk detail pengujian pada tiap desain akan ditampilkan pada Tabel 5.11.

Tabel 5.10 Rata-rata Hasil Uji Coba Kenaturalan Penggunaan Aplikasi

Desain	Rata-rata Nilai
Desain Tipe A	3.50
Desain Tipe B	3.50
Desain Tipe C	4.50

Tabel 5.11 Detail Penilaian Kenaturalan Penggunaan Aplikasi Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	3	4	4
Novita Retno P. L	4	5	5
Ibnu Prayogi	2	2	5
Relaci Apilia I	4	4	4
Mardiana Sekarsari	3	2	6
Hanif Sudira	5	4	3
TOTAL	21	21	27

Setelah dilakukan pengujian, pada Tabel 5.10 desain yang memiliki nilai kenaturalan tinggi adalah desain C dengan nilai 4.50, hal ini dikarenakan menurut pengguna desain ini hampir sama dalam memainkan drum aslinya. Kemudian desain yang memiliki kenaturalan kedua setelah desain tipe C adalah desain B dan A dengan nilai masing-masing 3.50 hal ini dikarenakan desain tampilan tidak sama dengan drum aslinya sehingga saat memainkannya menjadi kurang natural.

5.3.2.4 Hasil Uji Coba Kenyamanan Pengoperasian Aplikasi

Uji coba kenyamanan adalah pengujian apakah pengguna bisa merasakan nyaman ketika menggunakan aplikasi drum virtual. Hasil uji coba dipaparkan secara lengkap pada Tabel 5.12, sedangkan untuk detail pengujian pada tiap desain akan ditampilkan pada Tabel 5.13.

Tabel 5.12 Rata-Raja Hasil Uji Coba Kenyamanan Pengoperasian Aplikasi Setiap Desain

Desain	Rata-rata Nilai
Desain Tipe A	4.16
Desain Tipe B	3.66
Desain Tipe C	4.33

Tabel 5.13 Hasil Uji Coba Kenyamanan Pengoperasian Aplikasi Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	4	4	6
Novita Retno P. L	4	4	5
Ibnu Prayogi	3	3	4
Relaci Apilia I	5	4	4
Mardiana Sekarsari	4	3	4

Hanif Sudira	5	4	3
TOTAL	25	22	26

Setelah dilakukan pengujian, pada Tabel 5.12 terlihat bahwa desain yang memiliki kenyamanan adalah desain tipe C dengan nilai 4.33. Hal ini dikarenakan menurut pengguna desain C ukuran komponen drumnya besar beragam sehingga nyaman untuk dimainkan. Kemudian desain yang memiliki nilai kenyamanan kedua adalah desain tipe A, hal ini dikarenakan menurut pengguna desain tipe A dengan nilai 4.16 memiliki tampilan komponen drum lebih besar sehingga nyaman untuk dimainkan. Dan yang terakhir adalah desain tipe B dengan nilai 3.66. Desain tipe B memiliki kelemahan yaitu ukuran komponen drumnya yang kecil karena desain tata letaknya melingkar sehingga menurut pengguna desain tipe B kurang nyaman dimainkan.

5.4 Hasil Uji Coba Dependent Variable

Hasil uji coba *dependent variable* adalah hasil uji coba untuk mengukur secara objektif dari penggunaan aplikasi oleh pengguna. Hasil uji coba *dependent variable* disini berupa *task complete*, *task accuracy* dan *task error*.

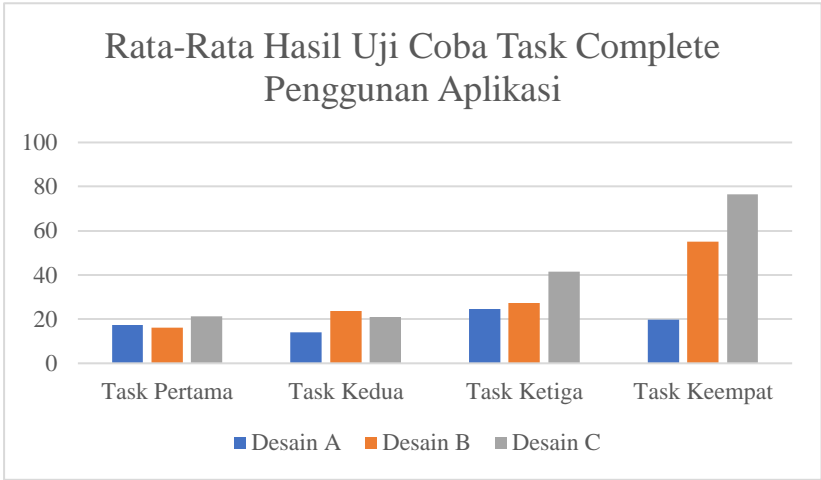
5.4.1 Hasil Uji Coba Task Complete Penggunaan Aplikasi

Uji coba *Task Complete* adalah pengguna memainkan aplikasi drum virtual sesuai naskah *drum beats* yang diberikan. Kemudian dihitung waktu penyelesaian pekerjaan tersebut. Waktu penyelesaian harus sesuai dengan tempo pada naskah *drum beats*. Berikut ini adalah hasil uji coba jumlah task complete pada Tabel 5.14 beserta grafik pada Grafik 5.1.

Tabel 5.14 Rata-Rata Hasil Uji Coba Task Complete

Task	Rata-Rata Task Complete Desain Aplikasi		
	A	B	C
Pertama	17.37	16.07	21.31

Kedua	14.04	23.82	20.92
Ketiga	24.58	27.29	41.58
Keempat	19.755	54.97	76.41
Rata-rata	18.93	30.53	40.05



Grafik 5.1 Rata-Rata Hasil Uji Coba *Task Complete* Penggunaan Aplikasi

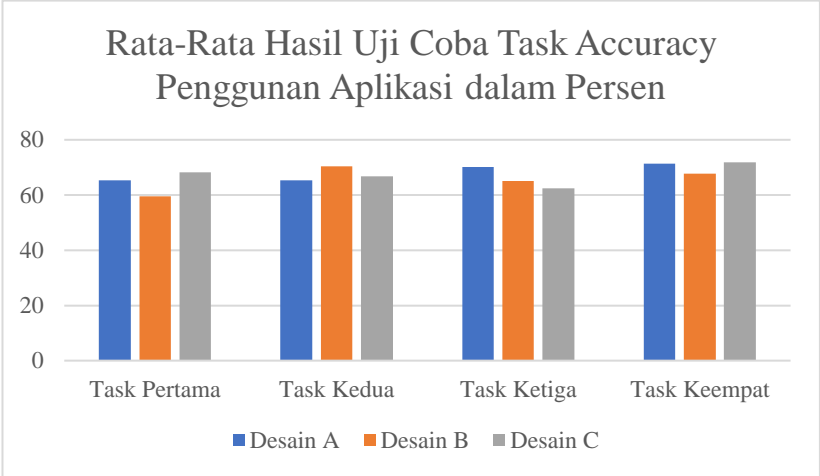
5.4.2 Hasil Uji Coba *Task Accuracy* Penggunaan Aplikasi

Hasil uji coba *task accuracy* adalah pengujian saat pengguna memainkan aplikasi drum virtual akan dihitung tingkat akurasi tempo pemukulan alat drum virtual tersebut. Berikut ini adalah hasil uji coba jumlah *task accuracy* pada Tabel 5.15 beserta grafik pada Grafik 5.2.

Tabel 5.15 Rata-Rata Hasil Uji Coba *Task Accuracy*

Task	Rata-Rata <i>Task Accuracy</i> Desain Aplikasi (%)		
	A	B	C
Pertama	65.31	59.43	68.27

Kedua	65.32	70.34	66.70
Ketiga	70.03	65.13	62.39
Keempat	71.28	67.83	71.83
Rata-rata	67.98	65.68	67.29



Grafik 5.2 Rata-Rata Hasil Uji Coba *Task Accuracy* Penggunaan Aplikasi dalam Persen

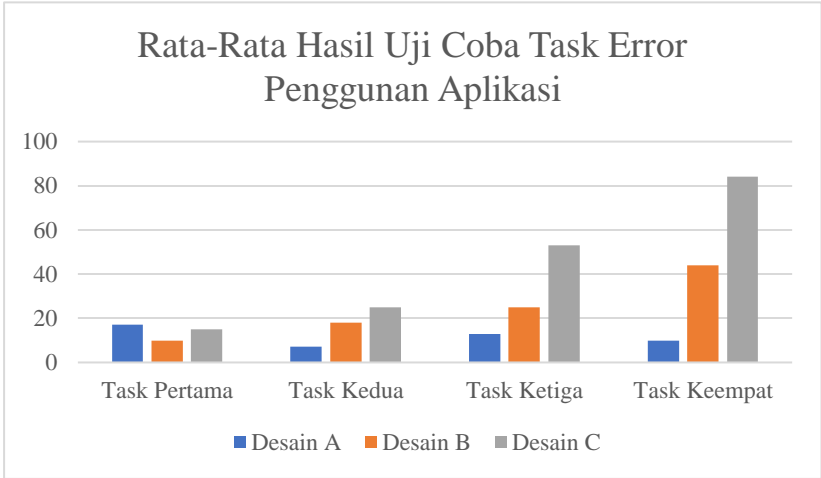
5.4.3 Hasil Uji Coba Jumlah *Task Error*

Hasil uji coba task error adalah saat pengguna memainkan aplikasi drum akan dilakukan seberapa sering pengguna melakukan kesalahan seperti salah memukul drum. Berikut ini adalah hasil uji coba jumlah task error pada Tabel 5.16 beserta grafik pada Grafik 5.3.

Tabel 5.16 Total Hasil Uji Coba *Task Error* Penggunaan Aplikasi

Task	Rata-Rata <i>Task Error</i>		
	A	B	C
Pertama	17	10	15

Kedua	7	18	25
Ketiga	13	25	53
Keempat	10	44	84
Total	47	97	177



Grafik 5.3 Rata-Rata Hasil Uji Coba *Task Error* Penggunaan Aplikasi

5.5 Evaluasi

Subbab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Dalam hal ini, evaluasi menunjukkan data rekapitulasi dari hasil pengujian fungsionalitas dan pengujian non-fungsionalitas yang telah dilakukan sebelumnya.

5.5.1 Evaluasi Pengujian Fungsionalitas

Evaluasi pengujian fungsionalitas dilakukan dengan menampilkan data rekapitulasi perangkat lunak yang telah dipaparkan pada subbab 5.2. Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Tabel 5.17. dari Data yang terdapat pada

tabel tersebut, diketahui bahwa aplikasi yang dibuat telah memenuhi kasus penggunaan yang telah ditentukan.

Tabel 5.17 Rekapitulasi hasil uji coba fungsionalitas

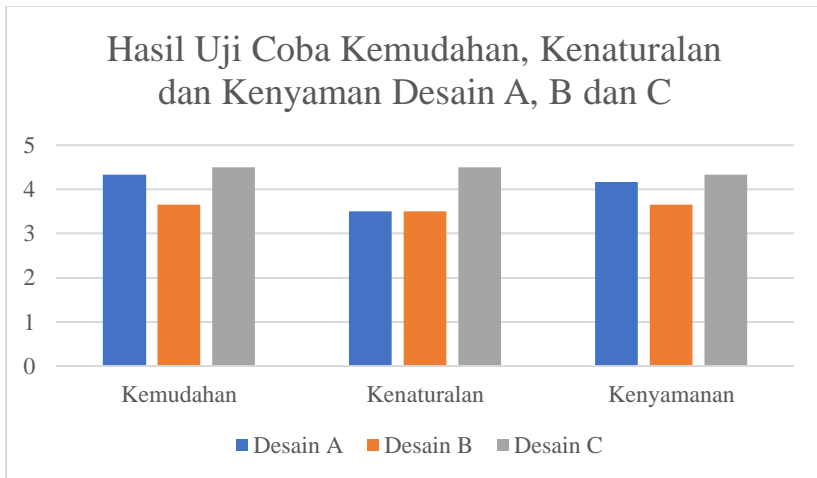
ID	Deskripsi		Hasil
UJ-UC-001	Uji Coba Memainkan aplikasi drum virtual	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
UJ-UC-002	Uji Coba pemindahan alat musik drum	Skenario 1	Berhasil
		Skenario 2	Berhasil
		Skenario 3	Berhasil
		Skenario 4	Berhasil
UJ-UC-003	Uji Coba Pengaturan Volume Suara	Skenario 1	Berhasil
		Skenario 2	Berhasil

5.5.2 Evaluasi Pengujian Non-Fungsionalitas

Evaluasi pengujian non-fungsionalitas dilakukan dengan menampilkan data rekapitulasi perangkat lunak yang telah dipaparkan pada subbab 5.3. Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Tabel 5.18 beserta grafik pada Grafik 5.4

Tabel 5.18 Rekapitulasi hasil uji coba non fungsionalitas

No	Deskripsi	Desain		
		A	B	C
1	Uji Coba Kemudahan Penggunaan Aplikasi	4.33	3.66	4.50
2	Uji Coba Kenaturalan Penggunaan Aplikasi	3.50	3.50	4.50
3	Uji Coba Kenyamanan Pengoperasian Aplikasi	4.16	3.66	4.33



Grafik 5.4 Hasil Uji Coba Non-fungsionalitas

Dari Grafik 5.4 dapat disimpulkan sebagai berikut ini:

1. Hasil uji coba kemudahan penggunaan penggunaan aplikasi, desain yang paling mudah digunakan adalah desain dengan tipe C (rating 4.5 dari 6) kemudian di urutan yang kedua adalah desain dengan tipe A (rating 4.33 dari 6) dan di urutan ketiga adalah desain tipe B (rating 3.66 dari 6).
2. Menurut hasil uji coba kenaturalan penggunaan aplikasi, desain yang paling natural adalah desain dengan tipe C (rating 4.5 dari 6) dan urutan kedua adalah desain dengan tipe B dan C masing (3.5 dari 6)
3. Menurut hasil uji coba kenyamanan pengoperasian aplikasi, desain yang paling nyaman adalah desain tipe C (rating 4.33 dari 6), kemudian di urutan yang kedua adalah desain tipe A (rating 4.16 dari 6) dan urutan yang ketiga adalah desain dengan tipe B (rating 3.66 dari 6).

5.5.3 Evaluasi Pengujian Dependent Variable

Evaluasi pengujian non-fungsionalitas dilakukan dengan menampilkan data rekapitulasi perangkat lunak yang telah dipaparkan pada subbab 5.4. Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada Tabel 5.18.

Tabel 5.19 Rekapitulasi hasil uji coba dependent variable

No	Deskripsi	Desain		
		A	B	C
1	Rata-rata uji coba <i>task complete</i> (dalam detik)	18.93	30.53	40.05
2	Rata-rata uji coba <i>task accuracy</i> (dalam persen)	67.98	65.68	67.29
3	Total uji coba <i>task error</i>	47	97	177

Berikut adalah penjabaran dari data rekapitulasi hasil uji coba dependent variable.

1. Menurut hasil uji coba *task complete*, desain yang paling cepat dikerjakan adalah Desain A sebesar 18.93 detik, kemudian yang kedua adalah Desain B sebesar 30.53 detik dan yang ketiga adalah Desain C sebesar 40.05 detik.
2. Menurut hasil uji coba *task accuracy*, desain yang paling akurasi adalah Desain A sebesar 67.98 persen, kemudian yang kedua adalah Desain C sebesar 67.29 persen dan yang terakhir adalah Desain B 65.68 persen.
3. Menurut hasil uji coba *task error*, desain yang paling sedikit tingkat kesalahan adalah Desain A sebesar 47 total kesalahan, kemudian yang kedua adalah Desain B sebesar 97 dan yang terakhir adalah Desain C sebesar 177.

(Halaman ini sengaja dikosongkan)

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat pula saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1. Kesimpulan

Dalam proses pengerjaan tugas akhir mulai dari tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Teknologi *finger tracking* Intel Realsense dapat diimplementasikan untuk pembuatan aplikasi drum virtual.
2. Setelah dilakukan pengujian dengan cara melakukan *experiment task* dan melakukan beberapa kuesioner desain yang baik untuk untuk pembuatan aplikasi drum virtual adalah desain tipe C sedangkan untuk hasil *dependent variable* seperti *task complete*, *task accuracy* dan *task error* adalah desain tipe A.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi dan uji coba yang telah dilakukan.

1. Peletakan (tata letak) setiap komponen satu dengan yang lainnya sebaiknya lebih diregangkan/dijauhkan agar tidak susah saat dimainkan dengan tangan kanan dan tangna kiri.
2. Terdapat desain drum pada desain drum tipe A yang kurang menarik, sehingga harus dibenahi lagi tampilannya agar lebih menarik lagi.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- [1] *Noname*, “4 Free Drum Apps For Windows 8.” [Diakses: 26-Des-2016].
- [2] L. Motion, “AirBeats.” [Daring]. Tersedia pada: <https://apps.leapmotion.com/apps/187>. [Diakses: 26-Des-2016].
- [3] Wagino, “Alat musik,” *Wikipedia bahasa Indonesia, ensiklopedia bebas*. 08-Jul-2016.
- [4] “CARA MEMBACA NOT BALOK PADA DRUM | MADUROCK.” [Daring]. Tersedia pada: <http://madurarock.blogspot.co.id/2012/05/cara-membaca-not-balok-pada-drum.html>. [Diakses: 26-Des-2016].
- [5] BeMyApp, “First steps with Intel® RealSense™ SDK by Xavier Hallade,” 12:28:15 UTC.
- [6] D. W. Seo, H. Kim, J. S. Kim, dan J. Y. Lee, “Hybrid reality-based user experience and evaluation of a context-aware smart home,” *Computers in Industry*, vol. 76, hal. 11–23, Feb 2016.
- [7] “Inside the Intel RealSense Gesture Camera | Chipworks.” [Daring]. Tersedia pada: <http://www.chipworks.com/about-chipworks/overview/blog/inside-the-intel-realsense-gesture-camera>. [Diakses: 25-Des-2016].
- [8] “INTEL REALSENSE.” [Daring]. Tersedia pada: <http://intelrealsense.bemyapp.com/>. [Diakses: 25-Des-2016].
- [9] “IR Sensor | What is an IR Sensor?” [Daring]. Tersedia pada: http://www.education.rec.ri.cmu.edu/content/electronics/boe/ir_sensor/1.html. [Diakses: 25-Des-2016].

- [10] “libgdx/gdx-realsense,” *GitHub*. [Daring]. Tersedia pada: <https://github.com/libgdx/gdx-realsense>. [Diakses: 25-Des-2016].
- [11] D. oleh irfan Fazri, “makalah seni musik.” .
- [12] “Not Balok Lengkap.” .
- [13] R. T. Singkoh, A. S. Lumenta, dan V. Tulenan, “Perancangan Game FPS (First Person Shooter) Police Personal Training,” *JURNAL TEKNIK ELEKTRO DAN KOMPUTER UNSRAT*, vol. 5, no. 1, hal. 28–34, 2016.
- [14] “Unity 3D download for Windows - Free Software Directory.” [Daring]. Tersedia pada: <https://bestwinsoft.com/developers/utilities/unity-3d>. [Diakses: 25-Des-2016].
- [15] “User Experience dan User Interface,” *UX Indonesia*, 13-Jun-2016. .
- [16] J. C. N. 12 dan 2012, “What is User Experience Design? Cooper.” [Daring]. Tersedia pada: <http://www.cooper.com/journal/2012/11/what-is-user-experience-design>. [Diakses: 25-Des-2016].

LAMPIRAN A


KUISIONER PENGGUNA

[illegible]

Gambar A. 1 Kuesioner Oleh Rahmat Rijal

Usability Test Script

Kuisloner: Desain Aplikasi B



1. Berapa rating immediate pengaprepan aplikasi dengan desain type B?

1
2
3
4
5
6

2. Berapa rating transferal pengaprepan aplikasi dengan desain type B?


1
2
3
4
5
6

3. Berapa rating kanyaprepan pengaprepan aplikasi dengan desain type B?

1
2
3
4
5
6

*Tentukan jawaban anda sesuai dengan perintah soal! Jawablah dengan jujur!

Terima kasih. Selamat siang.




What's next?

Use <https://github.com/Design-Mentors/Usability-Test-Script> to find more usability test scripts.

Usability Test Script

Kuisior Desain Aplikasi C



1. Berapa ringkasan kuantitas pengaplikasian aplikasi dengan desain tipe C?

1 2 3 4 5
2. Berapa ringkasan kuantitas pengaplikasian aplikasi dengan desain tipe C?

1 2 3 4 5
3. Berapa ringkasan kuantitas pengaplikasian aplikasi dengan desain tipe C?

1 2 3 4 5

Target: 100% benar

Penyelesaian: 100% benar

Waktu: 10 menit

100% What is next?
 Use Usability - Collaborative Design Method. Try the free demo available

Gambar A. 2 Kuesioner Oleh Novita Retno P.L

What's next?
 Use i490—i490 is the Collaboration Design Platform. Try it for free: www.collab.net

Gambar A. 3 Kuersio

What's next?
Join CIMA – Collaborative Design Platform. For the free www.cima.io

Penyusunan Oleh Ibnu Prayogi

Gambar A. 3 Kuersioner Oleh Ibnu Prayogi

Usability Test Script

Usability Test – Ijin untuk Merekam

Terima kasih untuk berpartisipasi dalam uji kegunaan Anda. Kita akan melakukan uji untuk mengetahui apa Anda dan teman-teman Anda memiliki masalah dalam menggunakan aplikasi ini.

Sebelumnya, perkenalkan diri Anda dan teman-teman Anda di tempat yang sudah disediakan.

Sebelumnya, beritahu kami bagaimana Anda akan melakukan.

Siapa nama Anda dan siapa nama teman-teman Anda? (jika ada)

Siapa nama Anda? Rafael Aprilia I

Siapa nama teman-teman Anda? Adi / Adi / Adi


Siapa nama teman-teman Anda? Adi / Adi / Adi

What's next?

Use Usability – Usability Design Platform. Try it for free. [Download Usability](#)

Usability Test Script

Kuesioner Desain Aplikasi B



1. Berapa rating kemudahan penggunaan aplikasi dengan desain tipe B? 1 2 3 4 5 6
2. Berapa rating kemudahan penggunaan aplikasi dengan desain tipe B? 1 2 3 4 5 6
3. Berapa rating kemudahan penggunaan aplikasi dengan desain tipe B? 1 2 3 4 5 6

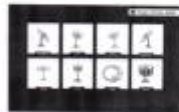
Jika tidak ada teman yang dapat Anda ajak, silakan lakukan uji sendiri saja.

What's next?

Use Usability – Usability Design Platform. Try it for free. [Download Usability](#)

Usability Test Script

Kuesioner Desain Aplikasi A



1. Berapa rating kemudahan penggunaan aplikasi dengan desain tipe A? 1 2 3 4 5 6
2. Berapa rating kemudahan penggunaan aplikasi dengan desain tipe A? 1 2 3 4 5 6
3. Berapa rating kemudahan penggunaan aplikasi dengan desain tipe A? 1 2 3 4 5 6


Tidak ada teman yang dapat Anda ajak, silakan lakukan uji sendiri saja.

What's next?

Use Usability – Usability Design Platform. Try it for free. [Download Usability](#)

Usability Test Script

Kuesioner Desain Aplikasi C



1. Berapa rating kemudahan penggunaan aplikasi dengan desain tipe C? 1 2 3 4 5 6
2. Berapa rating kemudahan penggunaan aplikasi dengan desain tipe C? 1 2 3 4 5 6
3. Berapa rating kemudahan penggunaan aplikasi dengan desain tipe C? 1 2 3 4 5 6

Tidak ada teman yang dapat Anda ajak, silakan lakukan uji sendiri saja.


What's next?

Use Usability – Usability Design Platform. Try it for free. [Download Usability](#)

Gambar A. 4 Kuesioner Oleh Relaci Aprilia I.

Uji Coba II

Kuisiener Desain Aplikasi II



- Berapa nilai rata-rata pengisian aplikasi dengan desain tipe A?


☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
- Berapa nilai rata-rata pengisian aplikasi dengan desain tipe B?

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
- Berapa nilai rata-rata pengisian aplikasi dengan desain tipe C?

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Disusun: Hengkyang, Nisrin, Alifhita

Kuisinter Desain Aplikasi C



3. Berapa rangkai kemutakhiran pengaplikasian aplikasi dengan desain tipe C?

(1) (2) (3) (4) (5)

4. Berapa rangkai kemutakhiran pengaplikasian aplikasi dengan desain tipe D?

(1) (2) (3) (4) (5)

5. Berapa rangkai kemutakhiran pengaplikasian aplikasi dengan desain tipe E?

(1) (2) (3) (4) (5)

(Catatan: Untuk menjawab kuis ini, klik tombol di bagian bawah. Setelah selesai klik tombol "Kirim Jawaban".)

WHAT'S NEW?
New arrivals | Collections | Fashion | Beauty | To Go | Free shipping worldwide

LAMPIRAN B

UJI COBA DEPENDENT VARIABEL

Tabel B. 1 Hasil Uji Coba *Task Complete* pada Task Pertama Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	6.81	8.19	7.82
Novita Retno P. L	4.84	8.46	7.9
Ibnu Prayogi	25.36	11.96	10.67
Relaci Apilia I	26.25	34.99	10.25
Mardiana Sekarsari	32.27	18.57	68.28
Hanif Sudira	8.71	14.25	22.99
TOTAL	104.24	96.42	127.91

Tabel B. 2 Hasil Uji Coba *Task Complete* pada Task Kedua Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	8.77	16.27	8.02
Novita Retno P. L	6.87	14.53	11.81
Ibnu Prayogi	12.41	39.81	25.21
Relaci Apilia I	21.3	23.54	17.91
Mardiana Sekarsari	21.51	21.51	24.06
Hanif Sudira	13.43	27.26	38.53
TOTAL	84.29	142.92	125.54

Tabel B. 3 Hasil Uji Coba *Task Complete* pada Task Ketiga Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	4.95	13.77	9.88
Novita Retno P. L	6.83	18.07	14.28
Ibnu Prayogi	10.01	59.19	6.54
Relaci Apilia I	10.1	58.35	95.02
Mardiana Sekarsari	12.81	7.14	59.57
Hanif Sudira	29.06	7.27	64.24
TOTAL	73.76	163.79	249.53

Tabel B. 4 Hasil Uji Coba *Task Complete* pada Task Keempat Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	9.19	11.37	11.37
Novita Retno P. L	10.81	19.77	18.02
Ibnu Prayogi	47.8	41.62	78.61
Relaci Apilia I	17.89	108.81	17.27
Mardiana Sekarsari	15.83	110.83	160.73
Hanif Sudira	17.01	37.46	172.47
TOTAL	118.53	329.86	458.47

Tabel B. 5 Hasil Uji Coba *Task Accuracy* pada Task Pertama Setiap Desain

Responden	<i>Task Accuracy</i> Desain Aplikasi (%)		
	A	B	C
Rahmat Rijal	74.51	60.96	68.6
Novita Retno P. L	59.38	70.59	66.6
Ibnu Prayogi	56.93	68.93	82.08
Relaci Apilia I	65.43	34.99	55.63
Mardiana Sekarsari	73.36	76.6	63.39
Hanif Sudira	62.26	44.53	73.37
TOTAL	391.87	356.6	409.67

Tabel B. 6 Hasil Uji Coba *Task Accuracy* pada Task Kedua Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	59.5	80	78.71
Novita Retno P. L	76.37	64.57	53.66
Ibnu Prayogi	58.48	61.63	76.49
Relaci Apilia I	53.29	74.64	78.12
Mardiana Sekarsari	81.27	81.27	54.51
Hanif Sudira	63.02	59.93	58.71
TOTAL	391.93	422.04	400.2

Tabel B. 7 Hasil Uji Coba *Task Accuracy* pada Task Ketiga Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	66.02	47.08	59.28
Novita Retno P. L	71.19	59.62	54.3
Ibnu Prayogi	64.01	65.15	56.27
Relaci Apilia I	62.88	74.46	60.47
Mardiana Sekarsari	78.62	80.18	77.64
Hanif Sudira	77.51	64.32	66.39
TOTAL	420.23	390.81	374.35

Tabel B. 8 Hasil Uji Coba *Task Accuracy* pada Task Keempat Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	91.95	73.37	73.37
Novita Retno P. L	68.64	73.29	70.25
Ibnu Prayogi	68.72	71.57	69.27
Relaci Apilia I	69.78	60.93	71.33
Mardiana Sekarsari	77.11	62.41	75.84
Hanif Sudira	51.5	65.44	70.97
TOTAL	427.7	407.01	431.03

Tabel B. 9 Hasil Uji Coba *Task Error* pada Task Pertama Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	0	0	0
Novita Retno P. L	0	1	0
Ibnu Prayogi	4	0	1
Relaci Apilia I	5	5	0
Mardiana Sekarsari	8	2	11
Hanif Sudira	0	2	3
TOTAL	17	10	15

Tabel B. 10 Hasil Uji Coba *Task Error* pada Task Kedua Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	0	1	0
Novita Retno P. L	0	4	3
Ibnu Prayogi	1	4	0
Relaci Apilia I	3	5	1
Mardiana Sekarsari	1	1	11
Hanif Sudira	2	3	10
TOTAL	7	18	25

Tabel B. 11 Hasil Uji Coba *Task Error* pada Task Ketiga Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	0	0	1
Novita Retno P. L	0	4	1
Ibnu Prayogi	4	7	0
Relaci Apilia I	1	8	16
Mardiana Sekarsari	4	6	28
Hanif Sudira	4	0	7
TOTAL	13	25	53

Tabel B. 12 Hasil Uji Coba *Task Error* pada Task Keempat Setiap Desain

Responden	Desain Aplikasi		
	A	B	C
Rahmat Rijal	0	0	0
Novita Retno P. L	0	2	2
Ibnu Prayogi	6	6	15
Relaci Apilia I	0	17	2
Mardiana Sekarsari	3	17	40
Hanif Sudira	1	2	25
TOTAL	10	44	84

BIODATA PENULIS



Penulis lahir di Probolinggo, 27 September 1994 merupakan anak kedua dari 3 bersaudara. Dalam menjalani pendidikan semasa hidup, penulis menempuh pendidikan di SDN SUKAPURA 1 Kab. Probolinggo, SMPN Sukapura 1, SMA Taruna Dra. Zulaeha, dan S1 Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS) pada rumpun Interaksi Grafika dan Seni (IGS).

Penulis berhasil mengembangkan *Software House* bernama Owline. Dari tahun 2014 saat berdirinya Owline sampai tahun 2016 penulis bersama anggota tim Owline lainnya berhasil mengimplementasi beberapa proyek pengembangan aplikasi baik website, android dan desain *UI/UX*. Sebagian besar klien merasa puas dengan hasil dari pembuatan proyek yang sudah dikerjakan. Pendidikan terakhir yang masih ditempuh adalah Sarjana Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember. Hingga saat ini penulis masih belajar perkembangan teknologi di masa yang akan datang. Selain itu menurutnya mahir di bidang Teknologi Informasi masih belum cukup untuk membawanya ke jalan sukses. Menurutnya sikap *leadership* perlu ditanamkan dalam kepribadian untuk memimpin Indonesia saat ini dan di masa yang akan datang.